CHAPTER 3

Strategies for Taxonomic and Functional Annotation of Metagenomes

Johan Bengtsson-Palme[*,†]

[*]*Department of Infectious Diseases, Institute of Biomedicine, The Sahlgrenska Academy, University of Gothenburg, Guldhedsgatan, Sweden* [†]*Center for Antibiotic Resistance research (CARe) at University of Gothenburg, Gothenburg, Sweden*

3.1 Introduction

Rapid advancements in DNA sequencing capacity have quickly transformed metagenomics in a radical fashion. While early approaches to both functional [1] and sequencing-based [2] metagenomics were mostly restricted by the throughput of the methods—in essence only allowing researchers to scratch the surfaces of microbial communities—the vast amounts of sequence data generated by today's sequencing platforms have largely changed the bottleneck towards computation, analysis, and interpretation capacity [3]. This poses new challenges for the field, and requires a reassessment of the strategies used to analyze metagenomic sequence data. However, the fast sequencing technology advances have been accompanied by a rich development of analysis tools, and the brisk expansion of available software options complicates the selection of the most appropriate ones. This chapter will broadly outline the available strategies for large-scale computational analysis of metagenomes and give examples of recent software developments for taxonomic and functional analysis of metagenomes. Those will be compared to well-established methods for performing certain metagenomic analysis tasks, such as taxonomic classification, functional analysis, metagenomic assembly, and comparisons between metagenomes. That said, the chapter will not aim to provide a complete review of all available software options—rather it will use examples in order to highlight useful exploration strategies for metagenomics. Finally, the chapter will discuss automated computational "pipelines" for analysis of metagenomic sequence data and highlight their pros and cons, in comparison to running the analyses individually.

3.2 Assessing Taxonomic Composition Using Metagenomic Data

Often, analysis of microbial communities using metagenomics is described as aiming to answer two core questions in ecology: who is there, and what are they doing? [4] These two questions

Metagenomics. https://doi.org/10.1016/B978-0-08-102268-9.00003-3

are tightly entwined, as the ecosystem functions that can be performed by a community are dictated by its inhabitants. Thus, it is of great importance to discern the taxonomic composition of a given microbial community, both because it provides interesting information on its own, but also because it serves as an important backdrop to analyses of biological functions. There are three general approaches to taxonomic classification (or taxonomic binning, as it is also referred to) of reads from shotgun metagenomic data: (i) mapping of sequences (usually quality filtered reads, but sometimes also longer assembled sequences) to a database of reference genomes; (ii) binning of sequences based on their composition of nucleotide k-mers; and (iii) classification of reads deriving from certain barcoding regions, particularly the small subunit (SSU) rRNA genes [5]. This section will discuss the advantages and problems with each of these methods, and outline in which situations each strategy may be most feasible (Fig. 3.1).

Mapping of reads (or other sequences) to genomes is a conceptually simple task; all reads from the sequenced metagenome are matched to a catalog of reference genomes, such as those present in the Human Microbiome Project [6] or NCBI RefSeq [7] databases, using sequence similarity searches. Generally, tools designed to allow only a modest degree of dissimilarity between the reads and the reference are used, such as BWA [8], Bowtie2 [9], or the STAR aligner [10], but also more sensitive tools like Diamond [11] can be used. In addition, there are a number of specialized tools that bundle the mapping step with postprocessing analysis to determine microbial community structure, including for example MEDUSA [12] and GOTTCHA [13]. Within the taxa covered by the reference database, read mapping approaches provide great resolution, even offering—at least in theory—discrimination between strains of the same species. The capacity for such fine-grained assignments grows with longer sequence lengths. However, as read lengths grow, the computational requirements for performing the read mapping also increases. Thus, the computational challenges for this type of analysis can be prohibitive, which particularly was the case with older software used for this purpose, such as BLAST [14] and BLAT [15]. However, with today's very efficient alignment algorithms computational speed often becomes a minor issue. If speed is a very important factor, a more computationally efficient alternative to read mapping is binning based on sequence composition. This strategy employs short nucleotide subsequences of a fixed length (k-mers) of, e.g., four or six bases, which are counted for each sequence. The sequence composition of k-mers is then compared to a reference database of k-mer compositions from known genomes, which must be prepared beforehand. There are several tools that employ this task, including Kraken [16], MetaCV [17] and the Ray assembler [18].

The downside of both these strategies is that they require a comprehensive reference database. In some types of environments, such as the human gut, the catalog of at least the more common inhabitants starts to be relatively complete [19], but in the vast majority of environments we only have reference sequences for some very common, scientifically interesting, or economically important species. As the predicted taxonomic assignments of each sequence will never become more accurate than permitted by the reference database, this is a strong limitation of using whole metagenome comparison methods outside of the

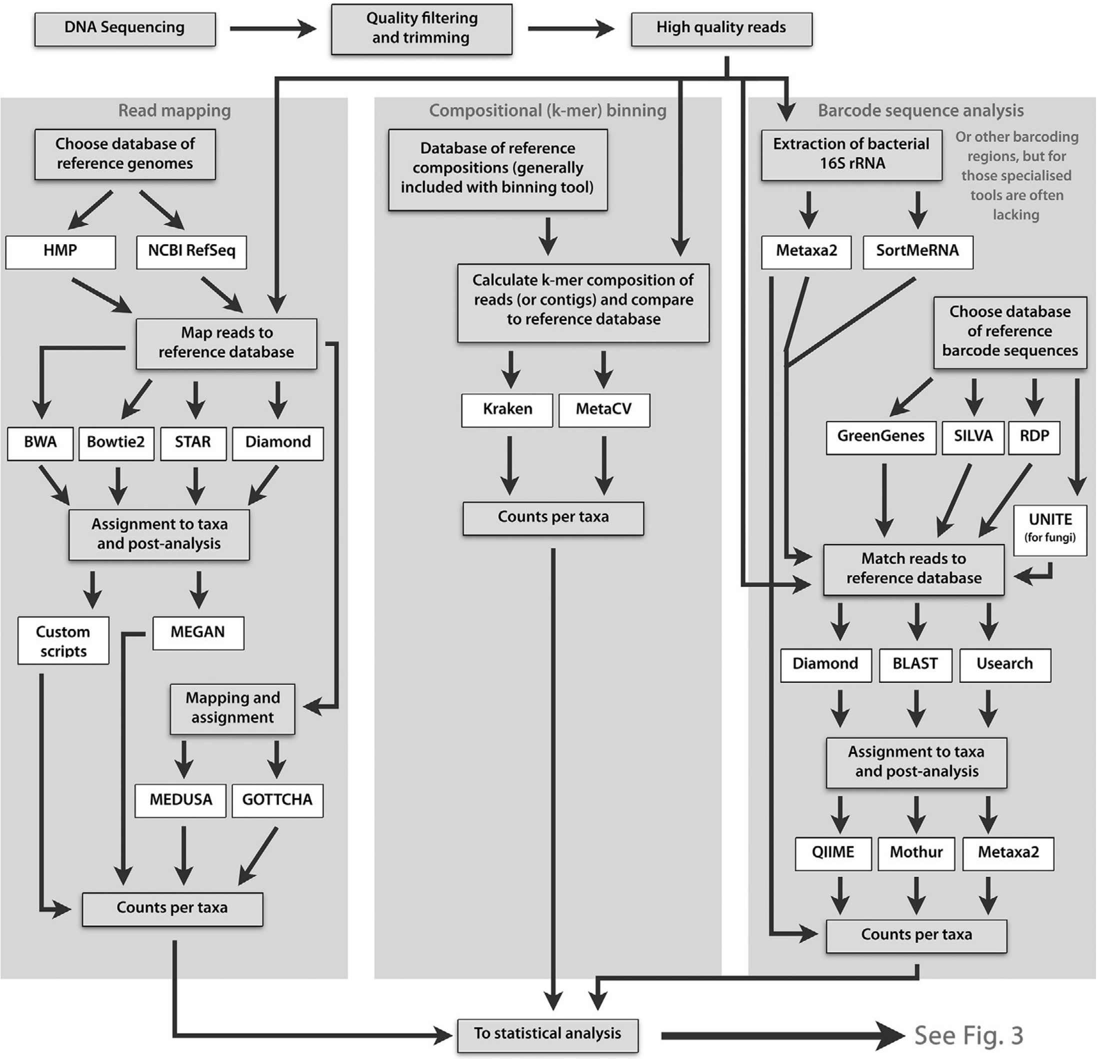
DNA Sequencing
Quality filtering and trimming
High quality reads
Read mapping
Choose database of reference genomes
HMP
NCBI RefSeq
Map reads to reference database
BWA
Bowtie2
STAR
Diamond
Assignment to taxa and post-analysis
Custom scripts
MEGAN
Mapping and assignment
MEDUSA
GOTTCHA
Counts per taxa
Compositional (k-mer) binning
Database of reference compositions (generally included with binning tool)
Calculate k-mer composition of reads (or contigs) and compare to reference database
Kraken
MetaCV
Counts per taxa
Barcode sequence analysis
Extraction of bacterial 16S rRNA
Or other barcoding regions, but for those specialised tools are often lacking
Metaxa2
SortMeRNA
Choose database of reference barcode sequences
GreenGenes
SILVA
RDP
UNITE (for fungi)
Match reads to reference database
Diamond
BLAST
Usearch
Assignment to taxa and post-analysis
QIIME
Mothur
Metaxa2
Counts per taxa
To statistical analysis
See Fig. 3

Fig. 3.1
Overview of the analysis workflow for taxonomic analysis of metagenomes. Three major strategies are outlined: (i) read mapping approaches, (ii) compositional binning, and (iii) analysis of barcoding regions. Note that while the compositional-based approaches are less complicated and time-consuming, those are also generally the ones with the lowest accuracy in terms of assigning reads to the correct taxa. The tools mentioned are suggestions, which are further discussed in the text. All these methods generate count data that can be used for further statistical analysis (see Fig. 3.3).

settings where the expected community inhabitants are well described, since the results will be greatly biased towards the already-sequenced community members (or their close relatives). On the contrary, there is a much larger catalog of reference sequences representing genetic regions used for DNA barcoding, i.e., the identification of species based on molecular methods [20]. Particularly for bacteria there are very large databases of the 16S SSU rRNA gene, which has been sequenced across a much greater range of species than completely sequenced genomes exist for. The largest such repositories for 16S rRNA genes are GreenGenes [21], the ribosomal database project [22], and SILVA [23]. For fungi, the UNITE database [24] represents a large controlled source of ITS barcoding sequences that can be used for the same purposes. To predict the taxonomic composition of microbial communities based on shotgun metagenomic data, software tools either compare all reads to a database of barcoding gene variants [25–28], or specifically identify reads from only such barcoding region prior to classification [29–32], often using heuristic approaches. Conceptually, the former approach is identical to the referenced-based comparison of full metagenomes, but uses a database only containing the barcoding region in question. This reference database is therefore much smaller in size, despite having better species coverage, which can dramatically lower computational time. On the other hand, using the prefiltration approach allows for using much more sensitive algorithms for the actual classification step, since the dataset size can be reduced a thousand-fold or sometimes more in the filtration step. Computationally efficient extraction of rRNA sequences from metagenomes can, e.g., be made using Metaxa2 [29] or SortMeRNA [31], of which Metaxa2 is also capable of performing the actual classification step with high accuracy. The output sequences from these tools can be classified using software commonly employed in molecular analysis of microbial communities, such as QIIME [33] and Mothur [34]. These tools include approaches that employ one (or combinations) of three different strategies [35]: (i) direct sequence alignments [25,36], (ii) comparisons of sequence compositions [37], or (iii) modeling of evolutionary divergence [29,38].

Depending on the aim of the study and the questions asked, different strategies for taxonomic analysis may be preferential. I would argue that the most central property of a taxonomic classification method is that it provides correct taxonomic affiliations rather than that it classifies every single input sequence to the species or genus level, although this view could be challenged depending on the circumstances and final use for the data. Different classification methods have recently been evaluated [29,39,40], and despite the fact that performance varies substantially even between software applying the same classification strategies, there are some more general conclusions that can be drawn. First, nearly all software tools in use today overestimated the number of species present in a mock community for which the number of taxa was known [39]. Second, methods based on the strategy of comparing sequences directly to a reference database generally have better precision than composition-based methods (i.e., those employing k-mers). This is—in general—true both for the barcoding-gene and the complete metagenome strategies [39,40]. It seems, however, that barcoding-gene approaches perform at least as well as mapping all reads from a metagenome [39], although a more comprehensive evaluation of this is still lacking. Among the barcoding approaches, there are also large performance variations [29], with Metaxa2 striking the best trade-off between precision and sensitivity, particularly at short read-lengths. Notably, Rtax [36] consistently shows high classification sensitivity (i.e., it classifies a large portion of the input sequences), but can at short read lengths make more than 50% misclassifications (i.e., poor precision) [29], a behavior that has also been observed when Rtax is used for other barcoding regions than the 16S rRNA [35].

Taken together, the evaluations of taxonomic classification methods that exist suggest that compositional-based methods, while being faster than strategies based on sequence similarity, should generally be avoided in all applications where correct taxonomic classifications are important. When making the choice between barcoding-based approaches and strategies applying comparisons of all reads in a metagenome to a genome reference database, the decisions should be based largely on database completeness. If the samples are derived from environments where most species can be expected to have been fully genome sequenced, full-scale comparisons make more sense as full genomes provide more fine-grained taxonomic resolution than barcoding genes do. However, even at a very short evolutionary distance from the fully sequenced genomes, the performance of full-metagenome methods becomes poor. Thus, if the community is suspected to contain many taxa without a fully sequenced close relative, the barcoding strategy is preferable, as it allows for more complete reference databases as well as lower identity thresholds in the comparisons to the references.

3.3 Metagenomic Assembly

As the length of the reads derived from shotgun metagenomic projects generally span from only 75 to a few hundred basepairs (depending on the sequencing platform), metagenomic assembly is often employed to forge these short reads together into longer continuous sequences called contigs. This practice is particularly useful for functional analysis of metagenomes, as it lends insights into the genetic context surrounding genes of interest, and also allows for more precise functional designation of predicted coding sequences. Furthermore, since the original reads can be mapped back to the contigs using software like Bowtie2 [9], BWA [8] or STAR [10], metagenomic assembly can also aid in estimating the abundance of certain features in the metagenome. However, metagenomic assembly is very computationally intensive and demands excessive amounts of computer memory (RAM), often requiring hundreds of gigabytes or even over a terabyte for large metagenomes [41,42]. Thus, metagenomic assembly should—at present—only be used when there is a clear benefit of doing so. For example, estimating the abundance of clearly defined genes (such as in the case of antibiotic resistance genes) by mapping reads to a reference database does not require—and is not improved by—performing an assembly prior to the analysis [43–46]. Similarly, taxonomy can also be inferred solely from raw read data. One of the reasons that this may be a more feasible approach than assembling the data and annotating the contigs is that the assembly process will bias the results towards the more common species in the metagenome, as those are more likely to contribute enough genetic material for stitching the reads together into contigs. Thus, DNA from less common organisms in the environment may be further undersampled artificially if only assembled data is analyzed. However, when the investigated gene families are less stringently defined, it is often more reasonable to do the analysis on assembled contigs, as that permits lowering the identity thresholds of the sequence comparison algorithms [46,47]. Thus, for full-scale overview annotation of a metagenome (see the next section), I would recommend assembling the metagenome and annotating the contigs, and then infer abundances using the contig read coverage information. Similarly, if the aim is to understand the genetic context of specific genes, metagenomic assembly is also, in principle, necessary [45,46]. For specific gene-mapping approaches and taxonomic analysis, on the other hand, I would suggest avoiding assembly as far as possible, as it can bias results and demands extensive computational infrastructure.

Early metagenomics projects generally relied on assembly software coupled to the sequencing platforms used. These assemblers were generally used on long-read data and based on the overlap-layout-consensus algorithm [48]. This worked well on smaller data sets, but quickly became an excessively time- and memory-consuming approach as metagenomic datasets grew in size, due to the fact that it requires all-to-all comparisons of all reads [49,50]. When short-read platforms generating vastly greater numbers of reads per experiment arrived, such as Illumina sequencing, these algorithms became nearly unusable, pushing the adoption of assembly algorithms employing de Bruijn graphs [51,52], which are less complex to build and traverse and thus make the assembly problem easier to solve [53]. This spawned a large number of multipurpose short-read assembler software programs which have been used for both genomic and metagenomic data, perhaps most commonly Velvet [54], ABySS [55], SOAPdenovo [56], Ray [57], and SPAdes [58]. Over time, some of these have also generated specialized versions adapted for metagenomic assembly [18,59]. These dedicated versions employ theoretically improved solutions for specific metagenomic assembly problems, but whether they offer any benefit in practice is poorly investigated. Some studies suggest, however, that the advantage of using, e.g., MetaVelvet rather than the regular Velvet version is negligible [60], and that in most cases the normal genomic assembly software does an equally good job.

As hinted previously, there are many assemblers to select from for metagenomic data, so how to make the choice? Generally, assembly aims to gain as long continuous stretches of DNA as possible from the reads, which has popularized the N50 metric as a benchmark for assembly quality. One arrives at the N50 value by including one contig after another (taking the longest first), until the selected contigs contain at least 50% of the assembled nucleotides. The N50 value then corresponds to the length of the shortest contig among those, i.e., the one selected last. Using the N50 value as a quality indicator is, however, flawed, particularly outside of single-genome assembly [61,62]. In theory, an assembler that just stapled all the input reads one after another would achieve a very high N50 metric, but the produced contig would not have any biological value. Indeed, benchmarking of how assemblers perform on reads from genomes where the true sequences are known show that the N50 is a poor predictor of quality in terms of producing assemblies corresponding to the real genomes [61,62]. These evaluations have also shown that many assemblers, including SOAPdenovo and Velvet, produce a large number of incorrect contigs, i.e., they merge sequences that do not belong together. On the other hand, ABySS and SPAdes seem less prone to this overmerging of reads [62], although not entirely void of it. That said, others have found that ABySS and Velvet perform rather similarly on short-read data from bacterial genomes [63]. Finally, it is also important that assemblers are scalable across large computer systems in order to maximize the use of available computing power. In this respect, parallelized assemblers such as ABySS and Ray overall excel in combining speed with accuracy of the final assemblies.

As mentioned, one problem with computational assembly is that it produces a certain degree of erroneous contigs. This problem may be even more pronounced for metagenomic assembly than it is for single genomes, as it is more complicated to tell assembly errors from, e.g., biological variations in a community context. Another problem related to large-scale assembly is that, with increasing dataset sizes, the computational demands also surge rapidly. To some degree this can be alleviated by employing strategies to reduce the complexity of the assembly, such as k-mer binning [42], filtering of low-coverage regions [64,65], or filtering of reads from high-coverage regions (digital normalization) [41]. An alternative strategy can be to divide samples into smaller datasets that are separately assembled and then subsequently merged [66], although this is likely to produce suboptimal assemblies. Furthermore, metagenomic assembly is complicated by the fact that identical genes, or regions of genes, may exist in many different species, which can create unsolvable situations for assembly software, with several possible connections between reads. This generally causes the assembler to fractionalize the assembly into several shorter contigs. Such splits cause severe problems in identifying the context around, e.g., antibiotic resistance genes, which are frequently very highly conserved and can exist in multiple genetic contexts [45]. Finally, metagenomic assembly is also limited by the coverage of the genomes in the microbial community, such that it requires each nucleotide to be covered by at least a few reads to be included in the assembly. Achieving such coverage across many different genomes is hard, and often impossible, causing the final assemblies to be biased towards the most common species and strains, limiting the interpretability of the results.

3.4 Detection and Quantification of Ecosystem Processes Using Metagenomics

An intriguing aspect of shotgun metagenomics compared to earlier approaches to microbial community ecology is that it not only allows inference of which taxa thrives in a particular environment, but also what kind of functional capacity these microorganisms carry. Metagenomics has been used both for broad functional overviews of communities in, e.g., the human gut [67,68], ocean surface water [69], and soil [70], but can also be applied to study specific research questions, such as antibiotic resistance selection in environments polluted with pharmaceuticals [45], or identification of alternative photosynthesis pathways [71]. Regardless of whether the questions asked are broad or more narrow, the choice of reference databases for functional annotation can greatly influence the final results of this process [72]. Poor choices of reference databases can essentially invalidate the main results of a study—regardless of how great the software tools used are—making it central to understand the content of the databases used for investigation of a given metagenome.

The databases used to make functional assignments to metagenomic sequences fall into two categories: those with broad scope, aiming to encompass the entire functional universe, and those with narrow scope covering one or a few specific biological processes. Examples of the latter processes for which specific databases exist include carbohydrate metabolism [73], nitrogen fixation [74], antibiotic resistance [75,76], metal tolerance [77], and bacterial virulence factors [78]. Broad databases are useful for getting an overview picture of the functionality of the studied community. For example, the Pfam database [79] can be used to assign predicted coding sequences to protein families. Most of these families are rather broad in scope, and even when they have clearly defined roles in the cell, they frequently act on a variety of substrates also within a single protein family (see, e.g., the ABC transporters). In contrast, specialized databases often contain manually curated sequence entries known to be associated with a precise function, such as export of a specific toxic compound. This means that if the aim is to, e.g., broadly understand detoxification in marine communities, Pfam families can provide a summary representation of whether such systems exist and their gross geographic distributions [47]. However, this approach will not provide much insight into the specificity of these enzymes and will for the same reasons not indicate if there are differential distributions of gene variants in different habitats. In many cases, the latter variation might be more important than the broad distribution of protein families, as many protein families that take part in central metabolic functions exist in a relatively stable number of copies across environments. Therefore, it is almost always desirable to use a database specialized for the research question whenever such a database exists. Sometimes, this means that the most feasible way of arriving at meaningful results involves constructing a new database enabling investigation of a particular question, based on the existing literature of described genes with verified functions. When selecting among different database options, it is important to consider the following aspects: (i) whether the sequences in the database are experimentally verified to perform the expected function; (ii) whether the annotation information (the metadata) is of high quality; and (iii) whether the database is comprehensive in terms of coverage of functions and taxa [72]. A database that does not fulfill these criteria may still be useful, but the results generated using it should be interpreted with these limitations in mind. If caution is not taken, particularly in these cases, there is a great risk that one may end up with erroneous or overstated conclusions.

Once a suitable database has been selected, the next step is to identify appropriate software tools for comparing the reads to the database (Fig. 3.2). While traditional tools like BLAST

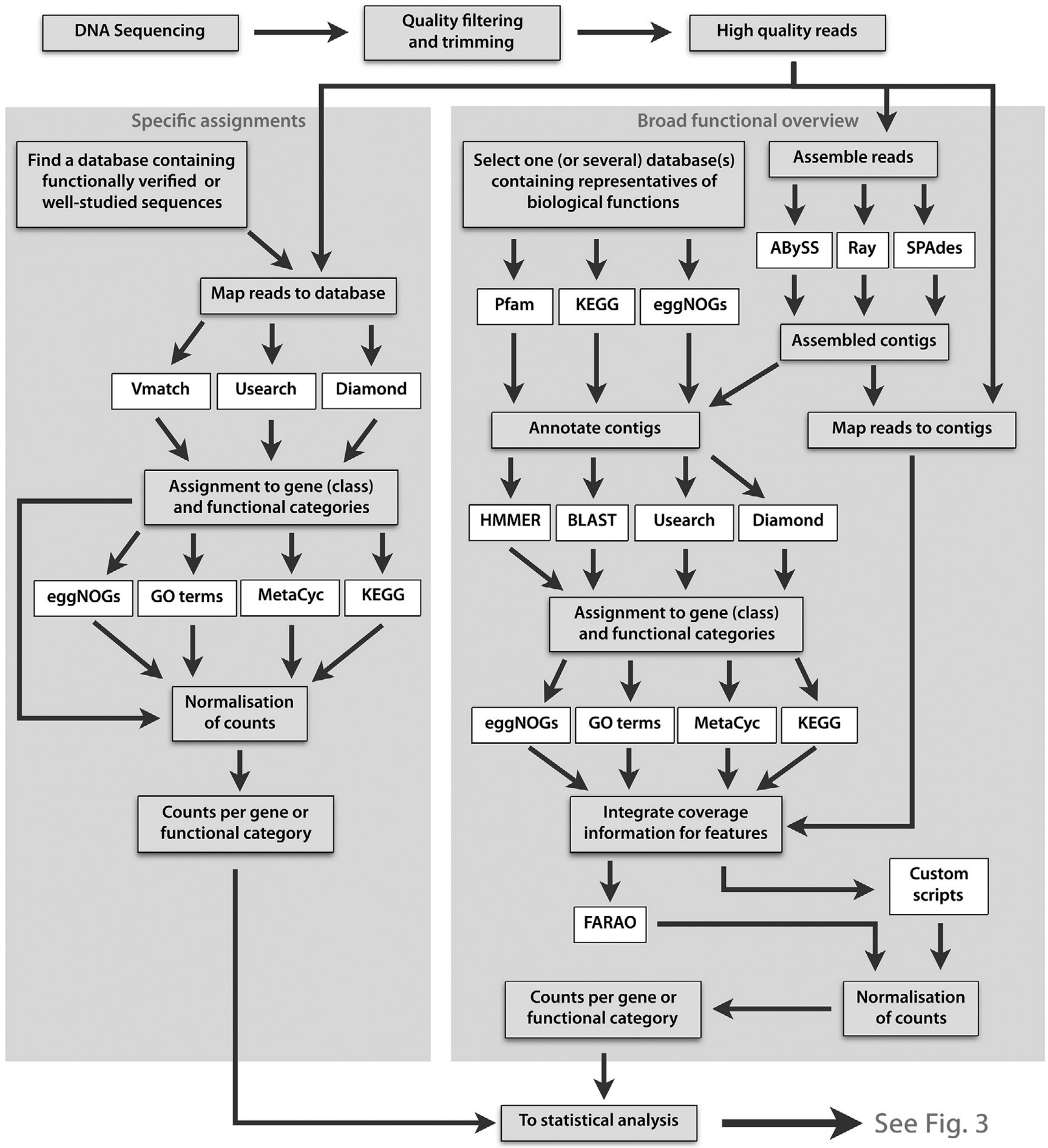
DNA Sequencing
Quality filtering and trimming
High quality reads
Specific assignments
Find a database containing functionally verified or well-studied sequences
Map reads to database
Vmatch
Usearch
Diamond
Assignment to gene (class) and functional categories
eggNOGs
GO terms
MetaCyc
KEGG
Normalisation of counts
Counts per gene or functional category
Broad functional overview
Select one (or several) database(s) containing representatives of biological functions
Assemble reads
ABySS
Ray
SPAdes
Pfam
KEGG
eggNOGs
Assembled contigs
Annotate contigs
Map reads to contigs
HMMER
BLAST
Usearch
Diamond
Assignment to gene (class) and functional categories
eggNOGs
GO terms
MetaCyc
KEGG
Integrate coverage information for features
Custom scripts
FARAO
Normalisation of counts
Counts per gene or functional category
To statistical analysis
See Fig. 3

Fig. 3.2
Overview of the analysis workflow for functional analysis of metagenomes. The suggested analysis steps are fundamentally different depending on whether a stricter, more specific (*left*), or a broad-encompassing annotation (*right*) strategy is desired. The tools and databases mentioned are suggestions, and particularly for the specific assignments a multitude of databases exist. All these methods generate count data that can be used for further statistical analysis (see Fig. 3.3).

[14] and BLAT [15] can be applied to search metagenomic sequences against database sequences, these approaches are most often too slow to be useful at the large scales imposed by the size of recent shotgun metagenomic projects. Instead, modern methods for comparing nucleotide sequences to protein databases employ optimized strategies that allow for 100-fold to 1000-fold speed increases. Some tools that are commonly used for this task are the Usearch [80] and Diamond [11] software applications. Occasionally, the versatile read mapper Vmatch [81] is also employed for searching metagenomic data for highly conserved sequences. It is important to understand, however, that all these tools trade sensitivity for speed, such that more distant homologs to the sequences in the reference database may not be detected using these tools, while they could have been identified using BLAST.

The preceding problem can be partially alleviated by assembling the reads before functional assignment, resulting in longer and fewer sequences to match to the databases. This first of all allows for higher probability of detecting distant homologs using the modern tools, but also brings the total dataset size down, which makes possible the use of more sensitive software, including BLAST. In addition, annotating the contigs resulting from an assembly instead of the raw reads can produce more reliable estimates of the coverage of, e.g., protein families, as the reads from each sample can be mapped back to the contigs to yield an estimate of the mean or median coverage of the feature in the metagenome. Tools for integrating different kinds of functional annotation with read coverage information exist, including FARAO [82] and Elviz [83]. Software like FARAO also becomes handy when combining annotation information based on, e.g., the Pfam database with the output of sequence-to-sequence based comparisons. Pfam represents each protein family with a hidden Markov model (HMM)—a statistical model describing the conserved features of each protein domain [79]. These models can be searched against sequence data using a software called HMMER [84]. In FARAO, the results of such searches can be added together with results of regular sequence similarity searches, as well as read coverage data generated by, e.g., Bowtie2 [9]. This allows for comparisons of annotations from different sources and estimation of the read coverage of individual Pfam families.

Regardless of the database and search strategy used, it is fundamental to choose appropriate cutoffs for scores, sequence identity and the length of the overlap between query and reference sequences. Failure to properly select identity thresholds can result in peculiar results, such as assignment of more than 4% of the genes in the human gut microbiome to antibiotic resistance [85]. Sadly, there is no easy way to generalize what is a good sequence identity cutoff to infer that two sequences have the same function. Even single amino acid substitutions can cause changes in protein functions [72], but at the same time proteins with less than 50% identity can still have the same cellular role [77]. In the quest for finding the correct threshold, I can only recommend that the researcher perform proper literature searches to identify what is reasonable within the particular group of proteins under study. In addition, regardless of what cutoff is chosen, the resulting data need to be interpreted with the identity or score threshold in mind—particularly when matching short reads to a database, as those will only cover part of the reference sequences. However, it might often be advisable to perform the similarity searches with less strict criteria, and then increase stringency when following up interesting findings, as this allows for filtering of high-identity matches without redoing the entire bioinformatic analysis.

It is also important to remember that the normalization strategy used can influence the results. Normalization of read counts aims to compensate for the fact that different sequencing libraries have different sizes (i.e., number of reads generated) and compositions (e.g., fraction of bacterial cells), as well as to compensate for foreseeable technical biases. The simplest way of compensating for sequencing depth is to simply divide each gene (or taxa or protein family) count by the number of reads generated for the corresponding sample. However, in many cases—if the study aims to investigate bacteria only—it is also interesting to compensate for, e.g., if there are a larger proportion of eukaryote or viral DNA in a particular sample. This can be done by instead dividing each count by some bacterial-specific marker gene. Often the 16S rRNA gene is used for this purpose. However, the copy number of the 16S gene varies between bacterial species and strains [86,87], which may introduce systematic biases into the analysis if there is a large community composition shift between two samples. Therefore, normalization strategies involving bacterial single-copy genes, such as the *rpoB* gene [47,88], have been proposed as better alternatives to 16S normalization. A strategy to further improve the accuracy of these estimates of bacterial abundance is to utilize several such single-copy (or stable-copy) genes for normalization [89,90]. Likely, the effects of copy number on the community level are fairly small, at least when communities get more complex. However, when comparing between different types of environments, where selective conditions may have driven different degrees of genome reduction due to, e.g., differences in nutrient availability [91], the influence of 16S copy number should be considered as a factor of bias. However, to what extent these normalization approaches actually influence real-world results is not well investigated. Another underinvestigated factor that may be important to compensate for is the length of the reference genes. In theory, longer genes would recruit more reads simply because they contribute more bases of DNA to the read pool than shorter genes do. If data is only compared between samples, gene length is not relevant to compensate for, but if the abundance levels within samples are compared, taking gene lengths into account becomes appropriate. It has, however, been debated whether length normalization matters in RNAseq applications [92–94], and similar lines of reasoning likely apply also to metagenomics.

Aside from selecting appropriate cutoffs, databases and normalization strategies, it is of course also crucial to sanity-check results. To this end, contextualizing findings related to specific genes in various ways is immensely helpful. Strategies to contextualize results include: (i) investigating the genetic neighborhood of the identified gene(s) using the contigs resulting from metagenomic assembly [46]; (ii) comparing several analysis methods to verify that they identify the same gene(s) as differential or important; (iii) exploring if other genes or taxa are co-enriched in similar environments, and if those genes or taxa have previously been reported to be connected to the same metabolic or ecosystem processes [95]; (iv) grouping genes by broader functions through, e.g., the terms in the Gene Ontology [96], and investigate if the entire process is co-enriched [45]; and (v) analyzing entire metabolic pathways, through, e.g., eggNOGs [97], MetaCyc [98] or the KEGG database [99], to verify that it is not a single gene in a pathway requiring several enzymes that has been found to be enriched [100,101]. While assembly and annotation methods have been discussed previously, and strategies to assess co-enrichment will be touched upon in following sections, network and pathway analysis is a wide subject area which cannot be sufficiently covered within this book chapter. Instead I recommend reading the review by De Filippo et al. on this subject [102].

3.5 Comparing Communities and Estimating Diversity

Once counts for genes or taxa have been obtained, it is most often desirable to compare different microbial communities to each other. A number of approaches exist for comparing communities, both at an overall level as well as for specific genes and taxa (Fig. 3.3). Strategies comparing the overall taxonomic or functional structures of communities often involve calculating a distance or dissimilarity measure between each pair of samples. These measures range from the relatively simplistic Euclidean distance to more sophisticated approaches designed specifically for comparing biological data, such as the Bray-Curtis dissimilarity [103], the Jaccard index [104] and the Raup-Crick index [105,106]. All these methods generate numbers describing how similar two communities are, and these numbers can be collected in a distance matrix (although strictly speaking many of these methods do not produce distances in a mathematical sense [107]). These distance matrices can then be used as input for statistical methods such as the Mantel test [108] for calculating the correlation between two distance matrices, Anosim [109] comparing the ranks of the distances within each group of samples to the ranks of the distances between different groups, and Permanova [110] which allows for performing multivariate ANOVA on the distance matrix and testing for differences between groups of samples through permutation. Permutation and resampling analysis of the individual counts can also be used to test for significant differences between samples or groups of samples, such as implemented in the Metaxa2 Diversity Tools [111]. Furthermore, there are several methods that can be used for data exploration and visualization, e.g., principal component analysis (PCA), principal coordinates analysis (PCoA), nonmetric multidimensional scaling (NMDS) and different clustering approaches. Clustering approaches are also highly useful for identifying groups of genes and taxa that are co-enriched or co-occur under similar conditions. These approaches all handle both functional and taxonomic data and many—but not all—are implemented in the Vegan R package for ecological analysis [112]. There is not enough space to properly address all of these methods here, but for the reader interested in more details, I would

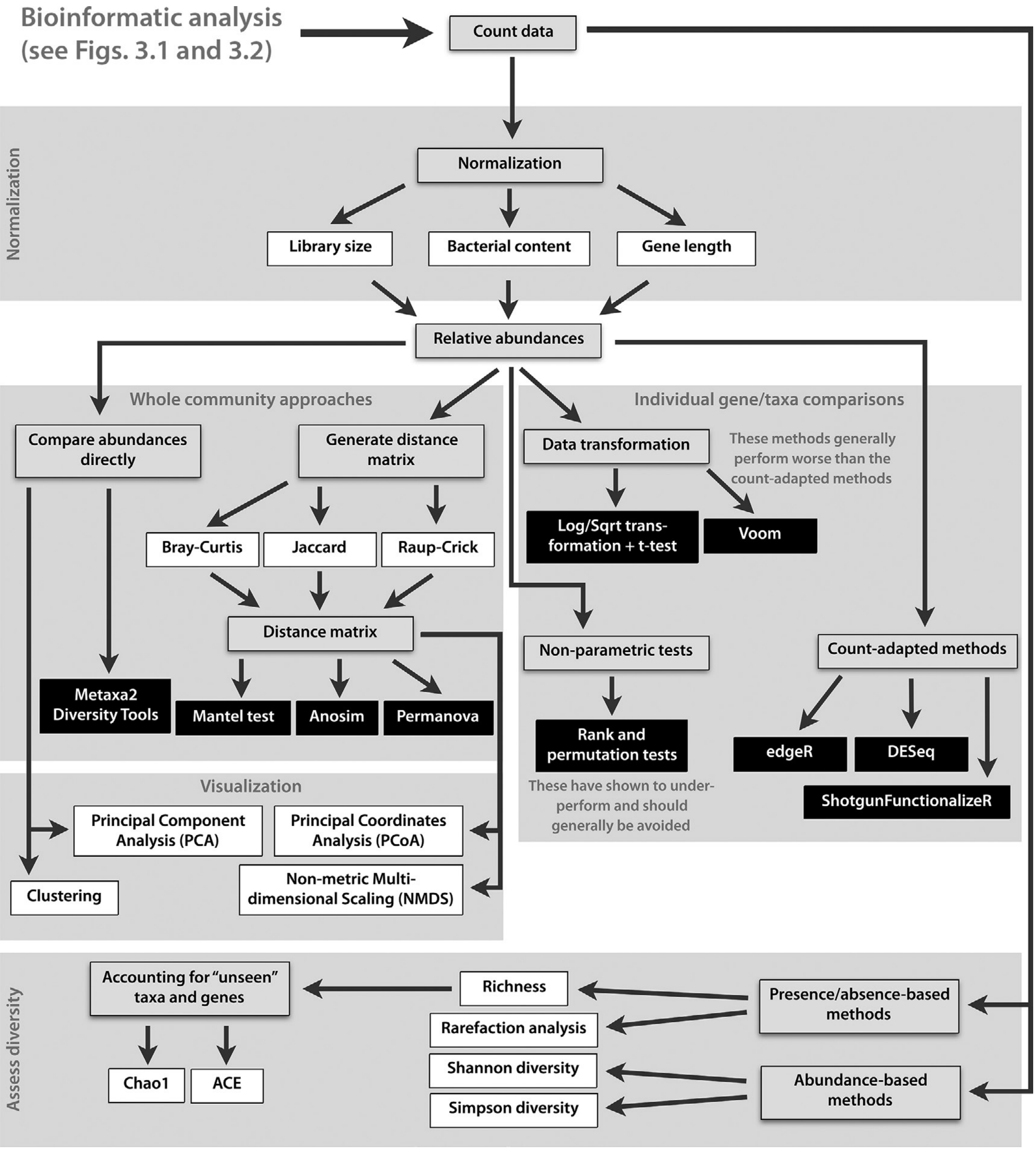
Bioinformatic analysis (see Figs. 3.1 and 3.2)
Count data
Normalization
Normalization
Library size
Bacterial content
Gene length
Relative abundances
Whole community approaches
Compare abundances directly
Generate distance matrix
Bray-Curtis
Jaccard
Raup-Crick
Distance matrix
Metaxa2 Diversity Tools
Mantel test
Anosim
Permanova
Individual gene/taxa comparisons
Data transformation
These methods generally perform worse than the count-adapted methods
Log/Sqrt trans-formation + t-test
Voom
Non-parametric tests
Rank and permutation tests
These have shown to under-perform and should generally be avoided
Count-adapted methods
edgeR
DESeq
ShotgunFunctionalizeR
Visualization
Principal Component Analysis (PCA)
Principal Coordinates Analysis (PCoA)
Clustering
Non-metric Multi-dimensional Scaling (NMDS)
Assess diversity
Accounting for "unseen" taxa and genes
Chao1
ACE
Richness
Rarefaction analysis
Shannon diversity
Simpson diversity
Presence/absence-based methods
Abundance-based methods

Fig. 3.3

Overview of the statistical analysis of metagenomes, starting with count data from taxonomic or functional analyses. Whole-community comparisons, comparisons between individual genes, taxa and functional groups, diversity assessments, as well as visualization strategies are briefly outlined and further described in the text.

recommend the recent overview by Paliy and Shankar, which well describes the different strategies available for these overall community comparisons [113].

In addition to these methods comparing the overall community structures between samples or groups of samples, it is often of interest to compare the relative abundances of specific genes or taxa (or entire biological systems) between different environments. However, statistical methods for handling metagenomic data face a multitude of problems. First of all, metagenomic data is very high-dimensional, i.e., there are many more observed genes than biological replicates. In addition, the variation between samples belonging to the same group is generally large, which creates a requirement for larger numbers of replicates to be able to detect statistically significant differences [114]. Unfortunately, DNA sequencing is still relatively expensive, meaning that one often has to strike an intricate balance between obtaining sufficient sequencing depth for quantification of genes in each separate sample and at the same time sequence enough replicate samples to detect significant differences. On top of this, the count data produced by metagenomic analysis is discrete, but many existing statistical tests assume that the data is continuous and normally distributed. To deal with these problems, a range of statistical approaches have emerged in the last decade [115]. Current statistical methods can be divided into three broad categories (Fig. 3.3): (i) standard tests on transformed counts, (ii) strategies assuming distributions accounting for the specific nature of discrete data, and (iii) nonparametric tests with fewer assumptions of how the underlying data is distributed.

One strategy to handle the statistics of metagenomics is to apply data transformations that enable the use of standard statistical tests, such as the Student's t-test and ANOVA. In addition, data transformations can also remove the tie between the mean and the variance inherent to count data. Common procedures used for this purpose are the square root transform and logarithm transforms. Logarithm-based transforms affect large values more than the square root transform does, which results in the logarithm-transformed data being less influenced by the most abundant genes. Transforming metagenomic data enables the use of readily available tools that have long been employed in, e.g., microarray analysis, such as Limma [116], as implemented in the Voom package [117]. However, the logarithm transforms also introduce problems when genes and taxa are not detected in some samples, as the logarithm of zero is undefined. This is commonly solved by adding what is referred to as a pseudocount to every single observation. However, the size of the pseudocount influences the effect sizes, particularly for samples and genes or taxa where overall counts are low. Furthermore, nondetects in metagenomic data are also problematic for another reason, namely that they can be produced in two different situations. Either they can represent a gene that is not present at all in the biological sample, or they may result from the fact that the sought-after gene is too rare to be detected given the sequencing depth used. The dual causes of zeros are not uniquely a problem for data transformation strategies, and are particularly troublesome in applications investigating the richness or diversity of taxa and genes.

The alternative to using data transformation is to employ statistical tests that make underlying assumptions better compatible with the nature of metagenomic count data, or do fewer assumptions on the data whatsoever. There already exists a number of statistical methods that are better designed for use in metagenomics projects, generally based on the Poisson or negative binomial distributions [114]. Such approaches have been implemented in, e.g., ShotgunFunctionalizeR [118], and two R packages originally developed for RNAseq; edgeR [119] and DESeq [120]. Methods that make few assumptions on how the data is distributed are referred to as nonparametric tests [121]. These methods, which are less sensitive to large variability within datasets and thus more robust to outliers, include permutation tests that resample the data instead of assuming its underlying distribution, and tests that calculate statistics based on the ranks of data points rather than their actual values, such as the Wilcoxon rank sum test [122].

Jonsson et al. recently performed an evaluation of different statistical approaches for metagenomic data and found that the number of replicates, the effect sizes, and the relative abundances of genes or taxa had very large effects on the outcomes of different methods [115]. Notably, no single method was shown to consistently perform better than the others—instead the best practice seems to be context-dependent. Nonetheless, the overall trend was that methods based on Poisson or negative binomial distributions performed better, particularly when group sizes were small. Specifically, the DESeq package and the overdispersed Poisson linear models were consistently among the highest performers. That said, even square root transformations followed by ordinary t-tests performed quite stably even for small group sizes. Finally, the study showed that one should avoid performing standard statistical tests without transforming the data, and that there is seldom any use of using nonparametric tests, as they routinely perform subpar to tests based on transformation or proper modeling of counts.

Finally, a question that has been debated in numerical ecology for decades is how to best measure diversity [123]. This question is usually discussed in the context of taxa, but is in essence equally applicable to functional systems or gene classes as well. Today, a range of diversity measures are used in community ecology, each carrying its advantages and disadvantages. One very basic measurement is to simply count the number of different taxa or gene types in a sample, resulting in what is referred to as the richness of the sample. A problem with this approach is that the measured richness is inherently coupled to the sampling effort—in this case corresponding to the size of the sequencing library. One approach to deal with this problem is to first normalize the counts for each gene to the size of its corresponding sample (the total number of reads), which results in gene frequencies that can be compared between samples. These normalized abundances can then be multiplied with the total size of the smallest sample, followed by counting only the number of entries with a value larger than or equal to one. While this reduces the dependency on library size, it at the same time introduces a bias towards the most abundant entities among all samples. Many authors have therefore suggested to instead use rarefaction methods, where the expected numbers of different taxa or gene types found at certain sampling efforts (sequencing depths) are calculated based on analytical calculations or resampling [111,124,125]. Regardless, these practices only account for the number of taxa found, and not at all their relative abundances. Several diversity indices also aim to capture this aspect of diversity, including the commonly used Shannon [126] and Simpson [127] indices. The Shannon index is in principle a measure of the entropy of the community—i.e., how far from evenly distributed the taxa or gene types are—while the Simpson index corresponds to the probability of sampling two individuals (or genes) from the same community belonging to the same taxa (or gene type). These indices are widely used, but the rationale for using a single index for measuring diversity has long been disputed [124]. It is also useful to recall that the obtained sample from the community only will contain a subset of the genes and taxa present in the full community. This means that the true richness of the sample is unknown, and that particularly information on nonabundant genes and taxa is poor or completely lacking. Similarly, only gene types with sufficient sequence similarity to some of the database reference sequences will be detected by metagenomics, biasing the analysis towards the genes that are already known. It may thus be useful to apply methods that try to estimate the unseen diversity as well. Again, several such measures exist, including the Chao1 [128] and ACE [129] estimators, as well as resampling methods [130]. However, these measures all fluctuate considerably as sample size changes and more data is added [131], and for the time being there is no truly good solution for accounting for unseen taxa and gene types. In my opinion, we are therefore probably better off comparing rarefaction curves, or the richness of detected genes and taxa at a certain sampling depth across different samples, and hope that those numbers correspond reasonably well to the true richness.

3.6 Computational "Pipelines" for Metagenomic Analyses

A tempting strategy for metagenomic analysis is to use one of the several "pipelines" for metagenomics that exists. These promise exploration of metagenomics data through simple point-and-click user interfaces, including execution of all computational analysis. Commonly used services that offer online computational pipelines for researchers are MG-RAST [132], METAREP [133] and EBI Metagenomics [134], but there are also a number of such tools that can be downloaded and used for offline analysis, including MOCAT2 [135], RAMMCAP [136], MetAMOS [137], and MEDUSA [12]. Among these, the results of MG-RAST analysis are probably those that are most often reported in literature.

The fact that these tools perform most or all of the analysis of the data is the strength and the weakness of these strategies. Automated pipeline approaches use standard sets of analysis tools and acceptance thresholds across all type of samples and gene classes, which allows for highly consistent comparisons but also may result in using very suboptimal criteria for some protein families, sample types, and research questions. That said, metagenomics analysis pipelines are excellent tools for providing a quick overview of the taxonomic and functional composition of a given community, or to swiftly evaluate how different metagenomes compare on a coarse level. However, for more precise analysis these pipeline solutions are often too blunt. This is underscored by the rather wide variability of functional categories used in the MG-RAST system, of which some are very broad and some quite narrow (Fig. 3.4). In addition, the cutoffs set in these pipelines will often be inappropriate for the group of genes studied, resulting in over- or underprediction of certain functional groups. This is well illustrated by research utilizing MG-RAST to define antibiotic resistance genes finding that 4% of the genes in human feces had an antibiotic resistance function [138], despite the fact that these estimates are more commonly pegged well below 1% [139,140]. Of course, this is partially also an effect of the fact that the databases underpinning services like MG-RAST are wide and general, and usually do not involve expert curation of the annotations, factors that are well known to introduce noise into sequence annotation [72]. Nevertheless, the power of pipelines to identify certain interesting aspects of the data that might have been overlooked using more specific databases and methods should not be neglected. Yet, such findings should always be followed up by more sensitive analysis strategies as well as literature studies, rather than directly reporting the results of a pipeline-based investigation.

There are, however, several reasons to be cautious about the results of pipelines, some of which make their use even for overview analysis complicated. The first reason is that users have less control of the sequence identity, alignment length and score thresholds used to infer annotations. Although this can often be controlled to some degree, many users simply use the predefined options, which are generally set too loosely to infer function with any substantial degree of certainty. Furthermore, the user has very restricted control of which databases to compare the metagenomic reads to, which primarily is a problem when aiming to investigate more specific biological aspects. But perhaps most importantly, these pipelines neglect the need for making informed decisions in every step of the analysis process—as they essentially allow bioinformatic analysis without a bioinformatician. This means that there is no one who can intervene in the process, and it is usually hard or impossible to check whether the intermediate analysis steps produce sensible output. Thus, software errors early in the computational pipeline can be masked by the subsequent analysis steps, and hence remain undiscovered by the researchers. Furthermore, it may be hard for nonbioinformaticians to tell whether a pipeline produces a reliable result or not (in the same way as it may be hard for bioinformaticians without a background in biology to tell whether their analysis makes biological sense). All of these problems can result in the downstream analyses being focused on the wrong questions, and may ignore important aspects of the data because too much trust is put into the results of the pipeline. In addition, the output produced by the computational pipelines invite researchers to largely design explorative

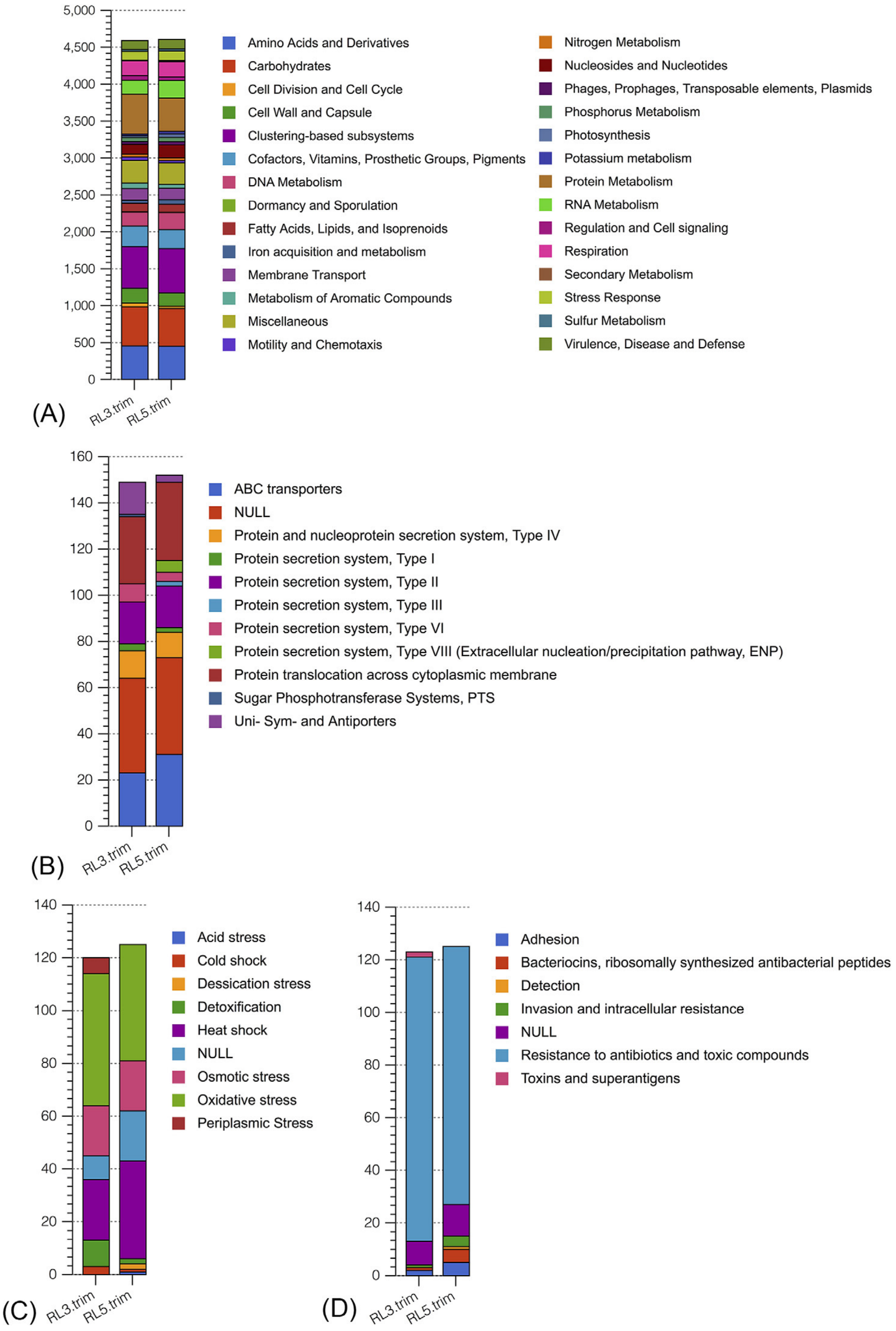

Fig. 3.4

See the legend in opposite page

studies, rather than setting up experiments driven by specific hypotheses. Such explorative study designs are often underpowered to properly detect significant differences between sample groups, and in this way pipeline analysis strategies may harm metagenomic analyses rather than helping them.

3.7 Conclusions and Outlook

The quick improvements in sequencing capacity over the last decade have created a requirement to continuously reassess the strategies used for metagenomic data analysis. Choosing the right tools is not an easy process, and there are several examples where the most popular tools have turned out to be less suited for particular tasks than many of the alternatives. While there does not seem to currently be any tools that obviously perform better in every situation, there are certain analysis strategies that clearly should be avoided. This chapter has attempted to highlight those, and at the same time suggest some robust and well-functioning software tools for taxonomic and functional analysis, metagenomic assembly and statistical comparisons. However, metagenomics is a field seeing rapid method development, and it is wise to revise the strategies and tools one uses every now and then. I would particularly advise researchers to actively search for papers evaluating the best available tools for assembly, annotation and statistical analysis of metagenomes. At the same time, many of those evaluation papers leave out a sizable number of the available tools [35,39,60,141,142], making it troublesome to compare between evaluation studies. Hopefully, future software evaluations will be more comprehensive and systematic, which would allow for better reassessments of the metagenomics toolbox.

Fig. 3.4, Cont'd
Functional groupings resulting from the use of metagenomics analysis "pipelines" can be highly variable, between broad-ranging and quite precise. In this figure, this is exemplified using output from MG-RAST subsystems analysis of two metagenomes (mgm4508983.3 (RL3) and mgm4508985.3 (RL5)). (A) Level 1 subsystems. Here, all described categories are relatively broad, and this figure generates useful overview information. However, at this level, most metagenomes from the same type of environment will usually have rather similar functional profiles, as many of these processes are necessary to maintain microbial life. (B) The "Membrane transport" level 2 subclass. Here, some functional groups are rather broad (such as "ABC transporters"), while some are quite narrow (e.g., "Protein secretion system, Type VIII (Extracellular nucleation/precipitation pathway, ENP)." Note also the totally noninformative category "NULL." The pattern of mixed broad and specific functional categories is repeated in the subclasses (C) "Stress Response" and (D) "Virulence, Disease and Defense." The variability in precision makes this information less useful for the researcher, and usually warrants more specific analyses using a database better curated for the purpose of the study.

References

[1] Handelsman J, Rondon MR, Brady SF, Clardy J, Goodman RM. Molecular biological access to the chemistry of unknown soil microbes: a new frontier for natural products. Chem Biol 1998;5:R245–9.

[2] Venter JC, et al. Environmental genome shotgun sequencing of the Sargasso Sea. Science 2004;304:66–74.

[3] Nayfach S, Pollard KS. Toward accurate and quantitative comparative metagenomics. Cell 2016;166:1103–16.

[4] Ma J, Prince A, Aagaard KM. Use of whole genome shotgun metagenomics: a practical guide for the microbiome-minded physician scientist. Semin Reprod Med 2014;32:5–13.

[5] Zepeda Mendoza ML, Sicheritz-Ponten T, Gilbert MTP. Environmental genes and genomes: understanding the differences and challenges in the approaches and software for their analyses. Brief Bioinform 2015;16:745–58.

[6] Human Microbiome Jumpstart Reference Strains Consortium, et al. A catalog of reference genomes from the human microbiome. Science 2010;328:994–9.

[7] O'Leary NA, et al. Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation. Nucleic Acids Res 2016;44:D733–45.

[8] Li H, Durbin R. Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics 2009;25:1754–60.

[9] Langmead B, Salzberg SL. Fast gapped-read alignment with Bowtie 2. Nat Methods 2012;9:357–9.

[10] Dobin A, et al. STAR: ultrafast universal RNA-seq aligner. Bioinformatics 2013;29:15–21.

[11] Buchfink B, Xie C, Huson DH. Fast and sensitive protein alignment using DIAMOND. Nat Methods 2015;12:59–60.

[12] Karlsson FH, Nookaew I, Nielsen J. Metagenomic data utilization and analysis (MEDUSA) and construction of a global gut microbial gene catalogue. PLoS Comput Biol 2014;10:e1003706.

[13] Freitas TAK, Li P-E, Scholz MB, Chain PSG. Accurate read-based metagenome characterization using a hierarchical suite of unique signatures. Nucleic Acids Res 2015;43:e69.

[14] Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. J Mol Biol 1990;215:403–10.

[15] Kent WJ. BLAT—the BLAST-like alignment tool. Genome Res 2002;12:656–64.

[16] Wood DE, Salzberg SL. Kraken: ultrafast metagenomic sequence classification using exact alignments. Genome Biol 2014;15:R46.

[17] Liu J, et al. Composition-based classification of short metagenomic sequences elucidates the landscapes of taxonomic and functional enrichment of microorganisms. Nucleic Acids Res 2013;41:e3.

[18] Boisvert S, Raymond F, Godzaridis E, Laviolette F, Corbeil J. Ray Meta: scalable de novo metagenome assembly and profiling. Genome Biol 2012;13:R122.

[19] Human Microbiome Project Consortium. Structure, function and diversity of the healthy human microbiome. Nature 2012;486:207–14.

[20] Taberlet P, Coissac E, Pompanon F, Brochmann C, Willerslev E. Towards next-generation biodiversity assessment using DNA metabarcoding. Mol Ecol 2012;21:2045–50.

[21] McDonald D, et al. An improved Greengenes taxonomy with explicit ranks for ecological and evolutionary analyses of bacteria and archaea. ISME J 2012;6:610–8.

[22] Cole JR, et al. The ribosomal database project (RDP-II): introducing myRDP space and quality controlled public data. Nucleic Acids Res 2007;35:D169–72.

[23] Yilmaz P, et al. The SILVA and 'All-species Living Tree Project (LTP)' taxonomic frameworks. Nucleic Acids Res 2014;42:D643–8.

[24] Kõljalg U, et al. Towards a unified paradigm for sequence-based identification of fungi. Mol Ecol 2013;22:5271–7.

[25] Huson DH, Mitra S, Ruscheweyh H-J, Weber N, Schuster SC. Integrative analysis of environmental sequences using MEGAN4. Genome Res 2011;21:1552–60.

[26] Segata N, et al. Metagenomic microbial community profiling using unique clade-specific marker genes. Nat Methods 2012;9:811–4.

[27] Liu B, Gibbons T, Ghodsi M, Treangen T, Pop M. Accurate and fast estimation of taxonomic profiles from metagenomic shotgun sequences. BMC Genomics 2011;(12 Suppl 2):S4.

[28] Darling AE, et al. PhyloSift: phylogenetic analysis of genomes and metagenomes. PeerJ 2014;2:e243.

[29] Bengtsson-Palme J, et al. Metaxa2: Improved identification and taxonomic classification of small and large subunit rRNA in metagenomic data. Mol Ecol Resour 2015;15:1403–14.

[30] Schmieder R, Lim YW, Edwards R. Identification and removal of ribosomal RNA sequences from metatranscriptomes. Bioinformatics 2012;28:433–5.

[31] Kopylova E, Noé L, Touzet H. SortMeRNA: fast and accurate filtering of ribosomal RNAs in metatranscriptomic data. Bioinformatics 2012;28:3211–7.

[32] Bengtsson J, et al. Metaxa: a software tool for automated detection and discrimination among ribosomal small subunit (12S/16S/18S) sequences of archaea, bacteria, eukaryotes, mitochondria, and chloroplasts in metagenomes and environmental sequencing datasets. Antonie Van Leeuwenhoek 2011;100:471–5.

[33] Caporaso JG, et al. QIIME allows analysis of high-throughput community sequencing data. Nat Methods 2010;7:335–6.

[34] Schloss PD, et al. Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities. Appl Environ Microbiol 2009;75:7537–41.

[35] Richardson RT, Bengtsson-Palme J, Johnson RM. Evaluating and optimizing the performance of software commonly used for the taxonomic classification of DNA metabarcoding sequence data. Mol Ecol Resour 2017;17:760–9.

[36] Soergel DAW, Dey N, Knight R, Brenner SE. Selection of primers for optimal taxonomic classification of environmental 16S rRNA gene sequences. ISME J 2012;6:1440–4.

[37] Wang Q, Garrity GM, Tiedje JM, Cole JR. Naive Bayesian classifier for rapid assignment of rRNA sequences into the new bacterial taxonomy. Appl Environ Microbiol 2007;73:5261–7.

[38] Munch K, Boomsma W, Huelsenbeck JP, Willerslev E, Nielsen R. Statistical assignment of DNA sequences using Bayesian phylogenetics. Syst Biol 2008;57:750–7.

[39] Peabody MA, Van Rossum T, Lo R, Brinkman FSL. Evaluation of shotgun metagenomics sequence classification methods using in silico and in vitro simulated communities. BMC Bioinform 2015;16:363.

[40] Bazinet AL, Cummings MP. A comparative evaluation of sequence classification programs. BMC Bioinform 2012;13:92.

[41] Howe AC, et al. Tackling soil diversity with the assembly of large, complex metagenomes. Proc Natl Acad Sci U S A 2014;111:4904–9.

[42] Pell J, et al. Scaling metagenome sequence assembly with probabilistic de Bruijn graphs. Proc Natl Acad Sci U S A 2012;109:13272–7.

[43] Lundström SV, et al. Minimal selective concentrations of tetracycline in complex aquatic bacterial biofilms. Sci Total Environ 2016;553:587–95.

[44] Pal C, Bengtsson-Palme J, Kristiansson E, Larsson DGJ. The structure and diversity of human, animal and environmental resistomes. Microbiome 2016;4:54.

[45] Bengtsson-Palme J, Boulund F, Fick J, Kristiansson E, Larsson DGJ. Shotgun metagenomics reveals a wide array of antibiotic resistance genes and mobile elements in a polluted lake in India. Front Microbiol 2014;5:648.

[46] Bengtsson-Palme J, et al. Elucidating selection processes for antibiotic resistance in sewage treatment plants using metagenomics. Sci Total Environ 2016;572:697–712.

[47] Bengtsson-Palme J, Alm Rosenblad M, Molin M, Blomberg A. Metagenomics reveals that detoxification systems are underrepresented in marine bacterial communities. BMC Genomics 2014;15:749.

[48] Staden R. A strategy of DNA sequencing employing computer programs. Nucleic Acids Res 1979;6:2601–10.

[49] Pop M. Genome assembly reborn: recent computational challenges. Brief Bioinform 2009;10:354–66.

[50] Miller JR, Koren S, Sutton G. Assembly algorithms for next-generation sequencing data. Genomics 2010;95:315–27.

[51] Idury RM, Waterman MS. A new algorithm for DNA sequence assembly. J Comput Biol 1995;2:291–306.

[52] Pevzner PA, Tang H, Waterman MS. An Eulerian path approach to DNA fragment assembly. Proc Natl Acad Sci U S A 2001;98:9748–53.

[53] Li Z, et al. Comparison of the two major classes of assembly algorithms: overlap-layout-consensus and de-Bruijn-graph. Brief Funct Genom 2012;11:25–37.

[54] Zerbino DR, Birney E. Velvet: algorithms for de novo short read assembly using de Bruijn graphs. Genome Res 2008;18:821–9.

[55] Simpson JT, et al. ABySS: a parallel assembler for short read sequence data. Genome Res 2009;19:1117–23.

[56] Li R, et al. De novo assembly of human genomes with massively parallel short read sequencing. Genome Res 2010;20:265–72.

[57] Boisvert S, Laviolette F, Corbeil J. Ray: simultaneous assembly of reads from a mix of high-throughput sequencing technologies. J Comput Biol 2010;17:1519–33.

[58] Bankevich A, et al. SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. J Comput Biol 2012;19:455–77.

[59] Namiki T, Hachiya T, Tanaka H, Sakakibara Y. MetaVelvet: an extension of Velvet assembler to de novo metagenome assembly from short sequence reads. Nucleic Acids Res 2012;40:e155.

[60] Vázquez-Castellanos JF, García-López R, Pérez-Brocal V, Pignatelli M, Moya A. Comparison of different assembly and annotation tools on analysis of simulated viral metagenomic communities in the gut. BMC Genomics 2014;15:37.

[61] Salzberg SL, et al. GAGE: A critical evaluation of genome assemblies and assembly algorithms. Genome Res 2012;22:557–67.

[62] Magoc T, et al. GAGE-B: an evaluation of genome assemblers for bacterial organisms. Bioinformatics 2013;29:1718–25.

[63] Narzisi G, Mishra B. Comparing de novo genome assembly: the long and short of it. PLoS One 2011;6:e19175.

[64] Hess M, et al. Metagenomic discovery of biomass-degrading genes and genomes from cow rumen. Science 2011;331:463–7.

[65] Mackelprang R, et al. Metagenomic analysis of a permafrost microbial community reveals a rapid response to thaw. Nature 2011;480:368–71.

[66] Scholz M, Lo C-C, Chain PSG. Improved assemblies using a source-agnostic pipeline for metagenomic assembly by merging (MeGAMerge) of contigs. Sci Rep 2014;4:6480.

[67] Qin J, et al. A human gut microbial gene catalogue established by metagenomic sequencing. Nature 2010;464:59–65.

[68] Arumugam M, et al. Enterotypes of the human gut microbiome. Nature 2011;473:174–80.

[69] Yooseph S, et al. The Sorcerer II Global Ocean Sampling expedition: expanding the universe of protein families. PLoS Biol 2007;5:e16.

[70] Charlop-Powers Z, et al. Urban park soil microbiomes are a rich reservoir of natural product biosynthetic diversity. Proc Natl Acad Sci U S A 2016;113:14811–6.

[71] Singh AH, Doerks T, Letunic I, Raes J, Bork P. Discovering functional novelty in metagenomes: examples from light-mediated processes. J Bacteriol 2009;191:32–41.

[72] Bengtsson-Palme J, et al. Strategies to improve usability and preserve accuracy in biological sequence databases. Proteomics 2016;16:2454–60.

[73] Cantarel BL, et al. The Carbohydrate-Active EnZymes database (CAZy): an expert resource for Glycogenomics. Nucleic Acids Res 2009;37:D233–8.

[74] Gaby JC, Buckley DH. A comprehensive aligned nifH gene database: a multipurpose tool for studies of nitrogen-fixing bacteria. Database (Oxford) 2014;2014:bau001.

[75] Jia B, et al. CARD 2017: expansion and model-centric curation of the comprehensive antibiotic resistance database. Nucleic Acids Res 2016;45:D566–73.

[76] Zankari E, et al. Identification of acquired antimicrobial resistance genes. J Antimicrob Chemother 2012;67:2640–4.

[77] Pal C, Bengtsson-Palme J, Rensing C, Kristiansson E, Larsson DGJ. BacMet: antibacterial biocide and metal resistance genes database. Nucleic Acids Res 2014;42:D737–43.

[78] Chen L, Zheng D, Liu B, Yang J, Jin Q. VFDB 2016: hierarchical and refined dataset for big data analysis—10 years on. Nucleic Acids Res 2016;44:D694–7.

[79] Finn RD, et al. Pfam: the protein families database. Nucleic Acids Res 2014;42:D222–30.

[80] Edgar RC. Search and clustering orders of magnitude faster than BLAST. Bioinformatics 2010;26:2460–1.

[81] Kurtz, S. The Vmatch large scale sequence analysis software. http://vmatch.de, 2010.

[82] Hammarén R, Pal C, Bengtsson-Palme J. FARAO: the flexible all-round annotation organizer. Bioinformatics 2016;32:3664–6.

[83] Cantor M, et al. Elviz—exploration of metagenome assemblies with an interactive visualization tool. BMC Bioinform 2015;16:130.

[84] Eddy SR. Accelerated profile HMM searches. PLoS Comput Biol 2011;7:e1002195.

[85] Nesme J, et al. Large-scale metagenomic-based study of antibiotic resistance in the environment. Curr Biol 2014;24:1096–100.

[86] Klappenbach JA, Dunbar JM, Schmidt TM. rRNA operon copy number reflects ecological strategies of bacteria. Appl Environ Microbiol 2000;66:1328–33.

[87] Větrovský T, Baldrian P. The variability of the 16S rRNA gene in bacterial genomes and its consequences for bacterial community analyses. PLoS One 2013;8:e57923.

[88] Dahllöf I, Baillie H, Kjelleberg S. rpoB-based microbial community analysis avoids limitations inherent in 16S rRNA gene intraspecies heterogeneity. Appl Environ Microbiol 2000;66:3376–80.

[89] Sunagawa S, et al. Metagenomic species profiling using universal phylogenetic marker genes. Nat Methods 2013;10:1196–9.

[90] Manor O, Borenstein E. MUSiCC: a marker genes based framework for metagenomic normalization and accurate profiling of gene abundances in the microbiome. Genome Biol 2015;16:53.

[91] Giovannoni SJ, Cameron Thrash J, Temperton B. Implications of streamlining theory for microbial ecology. ISME J 2014;8:1553–65.

[92] Rapaport F, et al. Comprehensive evaluation of differential gene expression analysis methods for RNA-seq data. Genome Biol 2013;14:R95.

[93] Dillies M-A, et al. A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis. Brief Bioinform 2013;14:671–83.

[94] Oshlack A, Wakefield MJ. Transcript length bias in RNA-seq data confounds systems biology. Biol Direct 2009;4:14.

[95] Faust K, et al. Microbial co-occurrence relationships in the human microbiome. PLoS Comput Biol 2012;8:e1002606.

[96] Ashburner M, et al. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nat Genet 2000;25:25–9.

[97] Huerta-Cepas J, et al. eggNOG 4.5: a hierarchical orthology framework with improved functional annotations for eukaryotic, prokaryotic and viral sequences. Nucleic Acids Res 2016;44:D286–93.

[98] Caspi R, et al. The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of Pathway/Genome Databases. Nucleic Acids Res 2014;42:D459–71.

[99] Kanehisa M, et al. Data, information, knowledge and principle: back to metabolism in KEGG. Nucleic Acids Res 2014;42:D199–205.

[100] Gianoulis TA, et al. Quantifying environmental adaptation of metabolic pathways in metagenomics. Proc Natl Acad Sci U S A 2009;106:1374–9.

[101] Sanli K, et al. Metagenomic sequencing of marine periphyton: taxonomic and functional insights into biofilm communities. Front Microbiol 2015;6.

[102] De Filippo C, Ramazzotti M, Fontana P, Cavalieri D. Bioinformatic approaches for functional annotation and pathway inference in metagenomics data. Brief Bioinform 2012;13:696–710.

[103] Bray JR, Curtis JT. An ordination of the upland forest communities of southern Wisconsin. Ecol Monogr 1957;27:325–49.

[104] Jaccard P. The distribution of the flora in the alpine zone. New Phytol 1912;11:37–50.

[105] Raup DM, Crick RE. Measurement of faunal similarity in paleontology. J Paleontol 1979;53:1213–7.

[106] Chase JM, Kraft N, Smith KG, Vellend M, Inouye BD. Using null models to disentangle variation in community dissimilarity from variation in α-diversity. Ecosphere 2011;2:24.

[107] Legendre P, Legendre LFJ. Numerical ecology. Elsevier; 2012.

[108] Mantel N. The detection of disease clustering and a generalized regression approach. Cancer Res 1967;27:209–20.

[109] Clarke KR. Non-parametric multivariate analyses of changes in community structure. Austral Ecol 1993;18:117–43.

[110] Anderson MJ. A new method for non-parametric multivariate analysis of variance. Austral Ecol 2001;26:32–46.

[111] Bengtsson-Palme J, Thorell K, Wurzbacher C, Sjöling Å, Nilsson RH. Metaxa2 Diversity Tools: Easing microbial community analysis with Metaxa2. Ecol Inform 2016;33:45–50.

[112] Oksanen, J. et al. vegan: community ecology package. http://cran.r-project.org/web/packages/vegan/index.html, 2017.

[113] Paliy O, Shankar V. Application of multivariate statistical techniques in microbial ecology. Mol Ecol 2016;25:1032–57.

[114] Jonsson V, Österlund T, Nerman O, Kristiansson E. Variability in metagenomic count data and its influence on the identification of differentially abundant genes. J Comput Biol 2017;24:311–26.

[115] Jonsson V, Österlund T, Nerman O, Kristiansson E. Statistical evaluation of methods for identification of differentially abundant genes in comparative metagenomics. BMC Genomics 2016;17:78.

[116] Smyth GK. Linear models and empirical bayes methods for assessing differential expression in microarray experiments. Stat Appl Genet Mol Biol 2004;3. Article3.

[117] Law CW, Chen Y, Shi W, Smyth GK. Voom: precision weights unlock linear model analysis tools for RNA-seq read counts. Genome Biol 2014;15:R29.

[118] Kristiansson E, Hugenholtz P, Dalevi D. ShotgunFunctionalizeR: an R-package for functional comparison of metagenomes. Bioinformatics 2009;25:2737–8.

[119] Robinson MD, McCarthy DJ, Smyth GK. edgeR: a bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics 2010;26:139–40.

[120] Anders S, Huber W. Differential expression analysis for sequence count data. Genome Biol 2010;11:R106.

[121] Schlenker E. Tips and tricks for successful application of statistical methods to biological data. Methods Mol Biol 2016;1366:271–85.

[122] Wilcoxon F. Individual comparisons by ranking methods. Biom Bull 1945;1:80–3.

[123] Magurran AE. Measuring biological diversity. Blackwell Science Ltd; 2004.

[124] Hurlbert SH. The nonconcept of species diversity: a critique and alternative parameters. Ecology 1971;52:577–86.

[125] Hughes JB, Hellmann JJ. The application of rarefaction techniques to molecular inventories of microbial diversity. Methods Enzymol 2005;397:292–308.

[126] Shannon C, Weaver W. The mathematical theory of communication. Urbana: University of Illinois Press; 1949.

[127] Simpson EH. Measurement of diversity. Nature 1949;163:688.

[128] Chao A. Nonparametric estimation of the number of classes in a population. Scand J Stat 1984;11:265–70.

[129] Chao A, Lee S-M. Estimating the number of classes via sample coverage. J Am Stat Assoc 1992;87:210–7.

[130] Colwell RK, Coddington JA. Estimating terrestrial biodiversity through extrapolation. Philos Trans R Soc Lond B Biol Sci 1994;345:101–18.

[131] Hughes JB, Hellmann JJ, Ricketts TH, Bohannan BJ. Counting the uncountable: statistical approaches to estimating microbial diversity. Appl Environ Microbiol 2001;67:4399–406.

[132] Keegan KP, Glass EM, Meyer F. MG-RAST, a metagenomics service for analysis of microbial community structure and function. Methods Mol Biol 2016;1399:207–33.

[133] Goll J, et al. METAREP: JCVI metagenomics reports—an open source tool for high-performance comparative metagenomics. Bioinformatics 2010;26:2631–2.

[134] Hunter S, et al. EBI metagenomics—a new resource for the analysis and archiving of metagenomic data. Nucleic Acids Res 2014;42:D600–6.

[135] Kultima JR, et al. MOCAT2: a metagenomic assembly, annotation and profiling framework. Bioinformatics 2016;32:2520–3.

[136] Li W. Analysis and comparison of very large metagenomes with fast clustering and functional annotation. BMC Bioinform 2009;10:359.

[137] Treangen TJ, et al. MetAMOS: a modular and open source metagenomic assembly and analysis pipeline. Genome Biol 2013;14:R2.

[138] Durso LM, Miller DN, Wienhold BJ. Distribution and quantification of antibiotic resistant genes and bacteria across agricultural and non-agricultural metagenomes. PLoS One 2012;7:e48325.

[139] Bengtsson-Palme J, et al. The human gut microbiome as a transporter of antibiotic resistance genes between continents. Antimicrob Agents Chemother 2015;59:6551–60.

[140] Hu Y, et al. Metagenome-wide analysis of antibiotic resistance genes in a large cohort of human gut microbiota. Nat Commun 2013;4:2151.

[141] Kerepesi C, Grolmusz V. Evaluating the quantitative capabilities of metagenomic analysis software. Curr Microbiol 2016;72:612–6.

[142] Charuvaka A, Rangwala H. Evaluation of short read metagenomic assembly. BMC Genomics 2011;(12 Suppl 2):S8.