SqueezeM, a fully automatic metagenomic analysis pipeline from reads to bins

Javier Tamames*, Fernando Puente-Sánchez

Departamento de Biología de Sistemas, Centro Nacional de Biotecnología, CSIC, C/Darwin 3, 28045 Cantoblanco, Madrid, Spain

Abstract

SqueezeM is a full automatic pipeline for metagenomics/metatranscriptomics, covering all steps of the analysis. In contrast to other available tools, SqueezeM includes multi-metagenome support allowing the co-assembly of related metagenomes and the retrieval of individual genomes via binning procedures. Thus, Squeeze M features several unique characteristics: Coassembly procedure with read mapping for estimation of the abundances of genes in each metagenome, or coassembly of unlimited number of metagenomes via merging of individual metagenomes. It also includes binning and bin checking, for retrieving individual genomes. Internal checks for the assembly and binning steps inform about the consistency of contigs and bins, allowing to spot potential chimeras. Metatranscriptomic support via mapping of cDNA reads against reference metagenomes.
The results are stored in a database, where they can be easily exported and shared, and can be inspected anywhere using a web interface.

Introduction

The improvement of sequencing technologies has allowed the generalization of metagenomic sequencing, which has become a standard procedure for analysing the structure and functionality of microbiomes. Many novel bioinformatic tools and approaches have been developed to deal with the huge numbers of short read sequences produced by a metagenomic experiment. Besides the simple overwhelming amount of data, currently in the range of terabases, a metagenomic analysis is a complex task involving several non-standardized steps, involving different software tools which results are often not directly integrable. Several pipelines have been created for automatizing the most of these analyses [1–6]. Nevertheless, they differ in capacities and, most importantly, in the way they perform annotations. A standard metagenomic pipeline would involve read curation, assembly, gene prediction, and functional and taxonomic annotation of the resulting genes. However, most of these platforms skip the assembly and gene prediction steps, and rely instead in the direct annotation of the short reads, thus avoiding the difficult step of assembly. Nevertheless, there are several drawbacks of working with raw reads: Since it is based on homology searches for millions of sequences against huge reference databases, it usually requires very large CPU usage. Especially for taxonomic

assignment, the reference database must be as complete as possible to minimize errors [7]; and importantly, the sequences are often too short to produce accurate assignments [8,9]

Assembly, on the contrary, is advisable because it can recover bigger fragments of genomes, often comprising many genes. Having the complete sequence of a gene plus its context makes it much more trustable regarding its functional and taxonomic assignment. The drawback of assembly are the formation of chimeras by misassembling parts of different genomes, and the inability to assemble some of the reads, especially the ones from low-abundance species. The fraction of unused reads depends of several factors, especially sequencing depth and microbiome diversity, but in many instances it is usually low (sometimes below 30%). Recently some tools have been developed to reassemble the portion of reads that were not assembled in first instance, increasing the performance of this step [10].

Besides, assembly is strongly advisable because it makes it possible the recovery of quasi-complete genomes via binning procedures. The retrieval of genomes is a big step ahead in the study of the microbiome since it allows to link functions to the individual taxa in it, thus providing a much accurate ecological description of the functioning of the community. It is possible, for instance, to determine the key members of the microbiome (involved in particularly important functions), to infer potential interactions between members, and to advance in the understanding of the effect of perturbations. The best strategy for binning is the co-assembly of related samples, when available. By comparing the abundance and composition of the contigs in different samples, it is possible to determine which contigs belong to the same member of the microbiome (these having similar oligonucleotide composition, similar abundances in individual samples, and a co-varying pattern between different samples).

SqueezeM is a full automatic pipeline for metagenomics/metatranscriptomics, covering all steps of the analysis. The current pipelines do not include support for coassembling and binning. At most they allow the import of the results to display binning information [2]. In contrast to other available tools, SqueezeM includes multi-metagenome support allowing the co-assembly of related metagenomes and the retrieval of individual genomes via binning procedures. Thus, Squeeze M features several unique characteristics:
1) Coassembly procedure with read mapping for estimation of the abundances of genes in each metagenome
2) Coassembly of unlimited number of metagenomes via merging of individual metagenomes
3) Includes binning and bin checking, for retrieving individual genomes
4) the results are stored in a database, where they can be easily exported and shared, and can be inspected anywhere using a web interface.
5) Internal checks for the assembly and binning steps inform about the consistency of contigs and bins, allowing to spot potential chimeras.
6) Metatranscriptomic support via mapping of cDNA reads against reference metagenomes

SqueezeM can be downloaded from https://github.com/jtamames/SqueezeM

Methods

SqueezeM can be run in three different modes, depending of the type of multi-metagenome support. These modes are (Figure 1):
-Sequential mode: All samples are treated individually and analysed sequentially. This mode does not include binning.
-Coassembly mode: Reads from all samples are pooled and a single assembly is performed. Then reads from individual samples are mapped to the coassembly to obtain gene abundances in each sample. Binning methods allow to obtain genome bins.
-Merged mode: if many samples are available, co-assembly will probably crash because of memory requirements. This mode allows the co-assembly of an unlimited number of samples, using a procedure similar to the one used by (ref protocol). Briefly, samples are assembled individually and the resulting contigs are merged in a single co-assembly (see below). Then the analysis proceeds as in the coassembly mode.

SqueezeM uses a combination of custom scripts and external software packages for the different steps of the analysis. A more detailed description of these steps follows:

1.1 Data preparation
SqueezeM only needs a configuration file indicating the metagenomic samples and the location of their corresponding sequence files. The user can also provide an optional metadata file. The program creates the appropriate directories and prepares the data for further step. It is important to remark that SqueezeM does not perform any quality filtering on sequence reads, since these treatments can vary greatly case by case.

1.2 Assembly
When assembling big metagenomic datasets, computing resources, especially memory usage, are critical. SqueezeM uses Megahit [11] as its reference assembler, since we find it has an optimal balance between performance and memory usage. SPAdes [12] is also supported, and additional assemblers can be added easily. In the merged mode, each metagenome will be assembled separately and the resulting contigs will be merged and joined, using cd-hit [13] for removing redundant contigs and minimus2 [14] for the merging.
The resulting set of contigs is filtered by length using prinseq [15], to discard short contigs if desired. In case of adding metatranscriptomic samples, these can be flagged for no assembly but will be considered in subsequent steps of the analysis.

1.3 Gene and rRNA prediction
This step performs a gene prediction on the resulting contigs, using the prodigal software [16], and looks for rRNAs using barrnap [17]. The program obtains ORFs and the corresponding aminoacid sequences of genes.

1.4 Homology searching
SqueezeM uses the Diamond software [18] for the comparison of the ORF sequences against several taxonomic and functional databases, because of its optimal computation speed while maintaining sensitivity. Currently, three different Diamond runs are used: against the GenBank nr database for taxonomic assignment, against the eggNOG database [19], and against the latest publicly available version of KEGG database [20]. These latter two are used for functional assignment in COGs and KEGG IDs, respectively.

1.5 Taxonomic and functional assignment of genes
Custom scripts are used for this step of the analysis. For taxonomic assignment, SqueezeM implements a fast LCA algorithm that looks for the last common ancestor of the selected hits for each ORFs (supplementary methods). Functional assignments in COGs and KEGG IDs can be done using the best hit, or a more sensitive approach considering the consistency of the selected hits (supplementary methods)

1.6 Taxonomic assignment of contigs and chimera check
The taxonomic assignment of individual ORFs is used to produce assignments for the contigs. A contig is annotated to the taxon to which most of its genes belong to (supplementary methods). This is the consensus annotation for the contig. The conditions for these assignments can be set by the user, and can accommodate misannotations of few genes, recent HGT events, etc.
A chimerism score is computed for each contig, indicating how many of the genes are discordant with the consensus (supplementary methods). Contigs with high chimerism could be the result of misassembles, and can be flagged to be excluded from the analysis.

1.7 Coverage and abundance estimation for genes and contigs
For estimating the abundance of genes and contigs, SqueezeM relies on the mapping of the original reads onto the metagenomic contigs resulting from the assembly. The software Bowtie2 [21] is used for this task, followed by Bedtools [22] extraction the raw number of reads mapping to each gene. Custom scripts are used to compute the average coverage for genes and contigs, and normalized RPKM values for gene abundance.
Metatranscriptomic reads that were excluded in the assembly are also used in this step, thus obtaining an estimation of the expression of the genes belonging to the metagenome and a direct comparison between metagenomic and metatranscriptomic abundances.
SqueezeM stops here if invoked in sequential mode. Co-assembled metagenomes allow further binning procedures for obtaining genomes.

1.8 Binning
From the contig coverage in different samples computed previously, squeezeM uses different binning methods to separate contigs putatively coming from the same organism. Currently, Maxbin [23] and Metabat2 [24] are supported. Using the mapping of reads to contigs previously obtained, SqueezeM calculates average coverage and RPKM values for the bins.

1.9 Taxonomic assignment of bins and chimera check

As for the annotation of contigs (step 1.6), SqueezeM generates a consensus taxonomic assignment for the bin. A bin is annotated to the taxon to which most of its contigs belong to, according to the consensus annotation for the contigs in it. As previously, a chimerism score is computed for each bin, indicating how many of the contigs are discordant with the consensus taxonomic assignment of the bin

1.10 Bin check
The goodness of the bins is estimated using the checkM software [25]. Briefly, CheckM provides indications of the completeness, contamination and strain heterogeneity of a bin by first creating a profile of single-copy, conserved genes for the provided taxon and then evaluating how many of these genes were found, and how many were single-copy. SqueezeM automatizes CheckM runs for each bin, using the consensus annotation for the bin as the suggested taxonomic origin.

1.11 Merging of results
Finally, the system merges all these results and generates several tables: 1) a genes table, with all the information regarding genes (taxonomy, function, contig and bin origin, abundance in samples, aminoacid sequence). 2) A contig table, gathering all data from contigs (taxonomy, bin origin, abundance in samples, chimerism), and 3) A bin table with everything related to the bins (taxonomy, completeness, contamination, abundance in samples, chimerism).

2. Database creation
These three tables and the optional metadata will be used in the creation of a mysql database for the easy inspection of the data derived from the analysis. The database includes a web-based user interface that allows the easy creation of queries, so that the user does not need to have any knowledge on databases to operate it (Figure 2). The interface allows queries on one table (Genes, contigs or bins) or in combinations of tables, allowing complex questions such as "Retrieve contigs having genes related to trehalose from Bacteroidetes that are more abundant than 5x coverage" or "Retrieve antibiotic resistance genes active in one condition but not in other". The resulting information can be exported to a table.

Results

For illustrating the use of the SqueezeM software, we analysed 32 metagenomic samples corresponding to gut microbiomes of Hadza and Italian subjects [26], using the three modes of analysis. Together, the total number of reads for all the metagenomes is 829.163.742. We used a 64-CPU computer cluster with 756 Gb RAM sited in the National Center for Biotechnology, Madrid, Spain. After discarding contigs below 200 bps, the total number of genes was 4.613.697, 2.401.848 and 2.230.717 for the sequential, merged and co-assembled modes, respectively. Notice that the number of genes is lower in the two latter modes that involve co-assembly since the genes that are present in more than one metagenome will be counted just once in the co-assemblage but more than one in the individual samples. A more accurate comparison is shown in Figure 3, where a gene in the co-assembly is assumed to be present in one

sample if it can recruit some reads from that sample. As co-assemblies create a much bigger reference collection of contigs than individual metagenomes alone, even genes belonging to low-abundance taxa and represented by a few reads can be identified by recruitment, while they will probably fail to assemble in the individual metagenome because of their low abundance. In other words, co-assembly will produce contigs and genes from abundant taxa in one or more samples, that can be used to identify the presence of the same genes in samples in which these original taxa are rare. Therefore, it allows to discover the presence of many more genes in each sample.

Table 1 shows the characteristics of the analysis. Even if the merged mode obtains more contigs and genes than the co-assembly mode, we can see that the number of putatively chimeric contigs is lower in the second. Chimeric contigs are detected when genes in the contig have been phylogenetically assigned to different taxa by the last common ancestor algorithm. Therefore, the co-assembly mode is more accurate than the merged mode, but the second has the advantage of being able to work with an almost unlimited number of metagenomes.

The results of binning have been analysed according the values of completeness and contamination provided by checkM (Table 2). Here also there is a difference between the merged and the co-assembly modes, with the first providing more bins but more incomplete, and the second giving bins of higher quality. Both modes are capable to obtain quasi-complete genomes for tens of species, and hundreds of less complete genomes.

Figure 4 shows the abundance distribution of bins in samples. Differences in the presence and abundance of genomes in samples can be clearly seen. The Italian subjects show a clear distinctive profile that make the clustering together (with the sole exception of the individual IT5, which was also an outlier in the original analysis based on functions) [26]. Hadza subjects fall in different groups corresponding to the presence of diverse species, which parallel the distinctions found using functional profiles. This information is directly retrieved from squeezeM results, and offers a very informative view of the genomic composition and differences between the samples.

Conclussions

SqueezeM is a highly versatile pipeline that allows to analyse an unlimited number of metagenomes or metatranscriptomes in a very straightforward way. All steps of the analysis are included, starting with the assembly, subsequent taxonomic/functional assignment of the resulting genes, abundance estimation and binning to obtain as much as possible of the genomes in the samples. SqueezeM is designed to run in moderately-sized computational infrastructures, relieving the burden of coassembling tens of metagenomes by using sequential metagenomic assembly and ulterior merging of resulting contigs.
The program includes several checks on the results, as the detection of possible chimeric contigs and bins, and the estimation of completion of the latter using the checkM software. Finally, results can be easily inspected and managed since SqueezeM includes a built-in mySQL database that can be

queried via a web-based interface. This allows the creation of complex queries in a very simple way.
SqueezeM will be further expanded by the creation of new tools allowing in-depth analyses of the functions and metabolic pathways represented in the samples.

References

1. Glass EM, Meyer F. The Metagenomics RAST Server: A Public Resource for the Automatic Phylogenetic and Functional Analysis of Metagenomes. Handb. Mol. Microb. Ecol. I Metagenomics Complement. Approaches 2011; 325–331
2. Eren a. M, Esen C, Quince C, et al. Anvi ' o : an advanced analysis and visualization platform for ' omics data. PeerJ 2015; 3:1–29
3. Arumugam M, Harrington ED, Foerstner KU, et al. SmashCommunity: A metagenomic annotation and analysis tool. Bioinformatics 2010; 26:2977–2978
4. Li W. Analysis and comparison of very large metagenomes with fast clustering and functional annotation. BMC Bioinformatics 2009; 10:359
5. Kim J, Kim MS, Koh AY, et al. FMAP: Functional Mapping and Analysis Pipeline for metagenomics and metatranscriptomics studies. BMC Bioinformatics 2016; 17:
6. Abubucker S, Segata N, Goll J, et al. Metabolic reconstruction for metagenomic data and its application to the human microbiome. PLoS Comput. Biol. 2012; 8:
7. Pignatelli M, Aparicio G, Blanquer I, et al. Metagenomics reveals our incomplete knowledge of global diversity. Bioinformatics 2008; 24:2124–2125
8. Wommack KE, Bhavsar J, Ravel J. Metagenomics: Read length matters. Appl. Environ. Microbiol. 2008; 74:1453–1463
9. Carr R, Borenstein E. Comparative analysis of functional metagenomic annotation and the mappability of short reads. PLoS One 2014; 9:e105776
10. Hitch TCA, Creevey CJ. Spherical: An iterative workflow for assembling metagenomic datasets. BMC Bioinformatics 2018; 19:
11. Li D, Liu CM, Luo R, et al. MEGAHIT: An ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. Bioinformatics 2015; 31:1674–1676
12. Bankevich A, Nurk S, Antipov D, et al. SPAdes: A New Genome Assembly Algorithm and Its Applications to Single-Cell Sequencing. J. Comput. Biol. 2012; 19:455–477
13. Fu L, Niu B, Zhu Z, et al. CD-HIT: Accelerated for clustering the next-generation sequencing data. Bioinformatics 2012; 28:3150–3152
14. Sommer DD, Delcher AL, Salzberg SL, et al. Minimus: A fast, lightweight genome assembler. BMC Bioinformatics 2007; 8:
15. Schmieder R, Edwards R. Quality control and preprocessing of metagenomic datasets. Bioinformatics 2011; 27:863–864
16. Hyatt D, Chen GL, LoCascio PF, et al. Prodigal: Prokaryotic gene recognition and translation initiation site identification. BMC Bioinformatics 2010; 11:
17. Seemann T. Prokka: Rapid prokaryotic genome annotation. Bioinformatics 2014; 30:2068–2069
18. Buchfink B, Xie C, Huson DH. Fast and sensitive protein alignment using

DIAMOND. Nat. Methods 2015; 12:59–60
19. Huerta-Cepas J, Szklarczyk D, Forslund K, 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–D293
20. Kanehisa M, Goto S. KEGG: Kyoto Encyclopaedia of Genes and Genomes. Nucl. Acids Res. 2000; 28:27–30
21. Langmead B, Salzberg SL. Fast gapped-read alignment with Bowtie 2. Nat. Methods 2012; 9:357–359
22. Quinlan AR, Hall IM. BEDTools: A flexible suite of utilities for comparing genomic features. Bioinformatics 2010; 26:841–842
23. Wu YW, Simmons BA, Singer SW. MaxBin 2.0: An automated binning algorithm to recover genomes from multiple metagenomic datasets. Bioinformatics 2015; 32:605–607
24. Kang DD, Froula J, Egan R, et al. MetaBAT, an efficient tool for accurately reconstructing single genomes from complex microbial communities. PeerJ 2015; 3:e1165
25. Parks DH, Imelfort M, Skennerton CT, et al. CheckM: Assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. Genome Res. 2015; 25:1043–1055
26. Rampelli S, Schnorr SL, Consolandi C, et al. Metagenome Sequencing of the Hadza Hunter-Gatherer Gut Microbiota. Curr. Biol. 2015; 25:1682–1693

Acknowledgements

This research was funded by project CTM2016-80095-C2-1-R, Ministerio de Economía y Competitividad, Spain.

	Merged mode	Co-assembly mode	Sequential mode
Number of contigs	893438	983350	2478560
N50	3900	2357	2854
Contigs with phylum annotation	719098 (80.4%)	759903 (77.2%)	1951445 (78.7%)
Contigs with chimerism>0	6626 (0.7%)	3772 (0.4%)	7588 (0.3%)
Highly chimeric contigs (chimerism>0.25)	4496 (0.5%)	2993 (0.3%)	5433 (0.2%)
Number of genes	2401848	2230717	4613697
Total bins	563	423	N/A
Bins >90% complete	120	115	N/A
Bins >50% complete	359	192	N/A
High-quality bins (>90% complete, <10% contam)	50	67	N/A
Good quality bins >75% complete, <10% contam)	82	112	N/A

Table 1: Statistics on contigs and bins. Binning statistics refer to MaxBin results.

Taxa	Size (bp)	Completeness	Contamination
o:Clostridiales	3.098.646	99.53%	3.57%
g:Bacteroides	3.521.779	99.45%	0.18%
o:Aeromonadales	2.579.625	99.16%	0.69%
g:Akkermansia	3.031.328	98.94%	4.31%
g:Treponema	2.879.091	98.62%	4.78%

g: Prevotella	3.354.102	97.51%	0.47%
s: *Escherichia coli*	4.710.119	97.49%	6.01%
g: Bifidobacterium	2.266.937	97.40%	3.70%
g: Megasphaera	2.490.127	96.93%	2.88%
s: *Succinatimonas sp.*	2.250.348	96.61%	4.22%
g: Parabacteroides	4.558.677	96.57%	3.18%
s: *Alistipes putredinis*	2.258.860	95.28%	5.77%
s: *Oscillibacter sp.*	1.801.182	95.11%	4.20%
s: *Bacteroides sp.*	3.901.726	95.08%	7.76%

Table 2: Example of some relevant high-quality bins (>90% completion, <10% contamination) obtained by the co-assembly mode of Hadza metagenomes

Figures

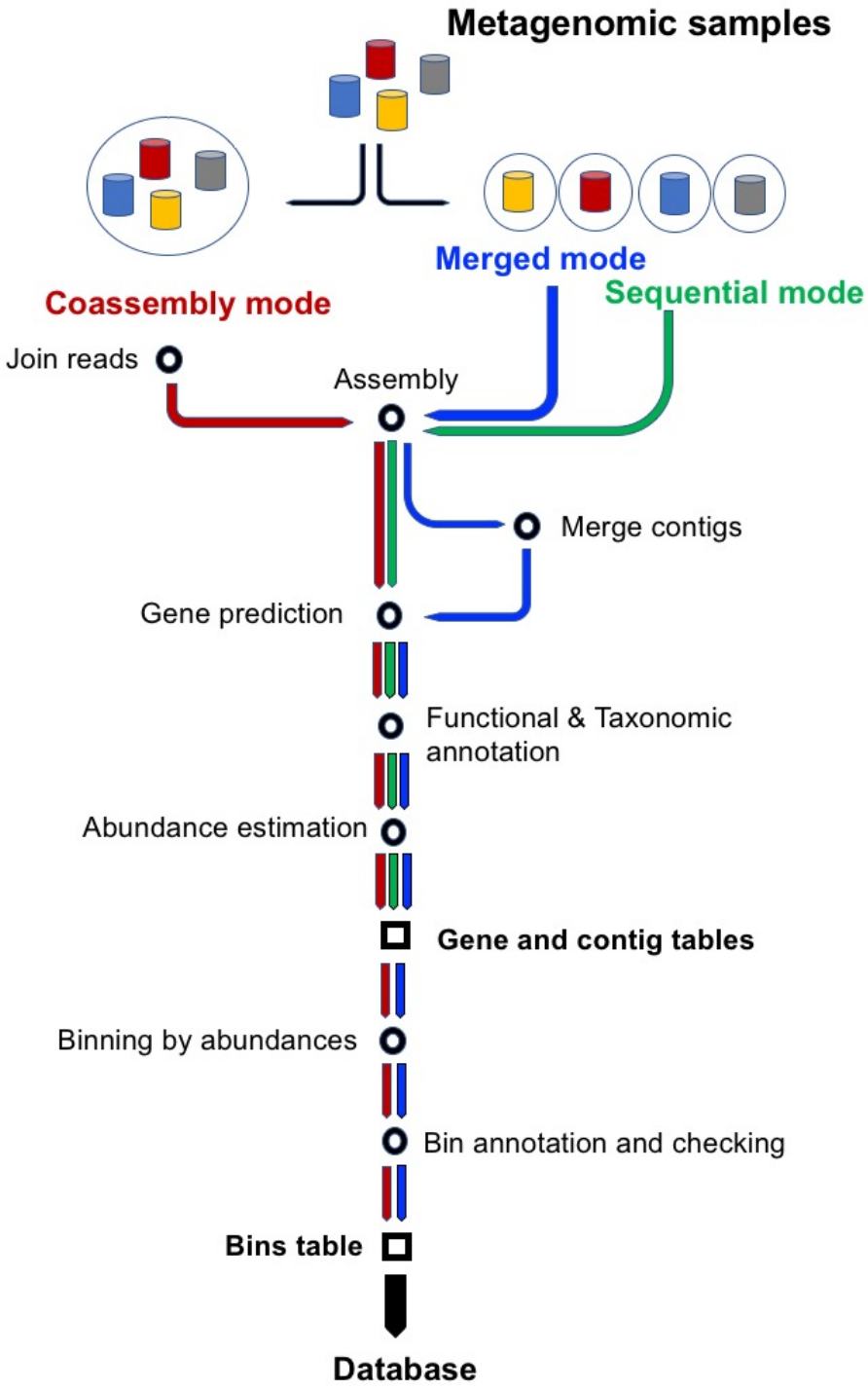

Figure 1: Description of the three modes of operation of SqueezeM

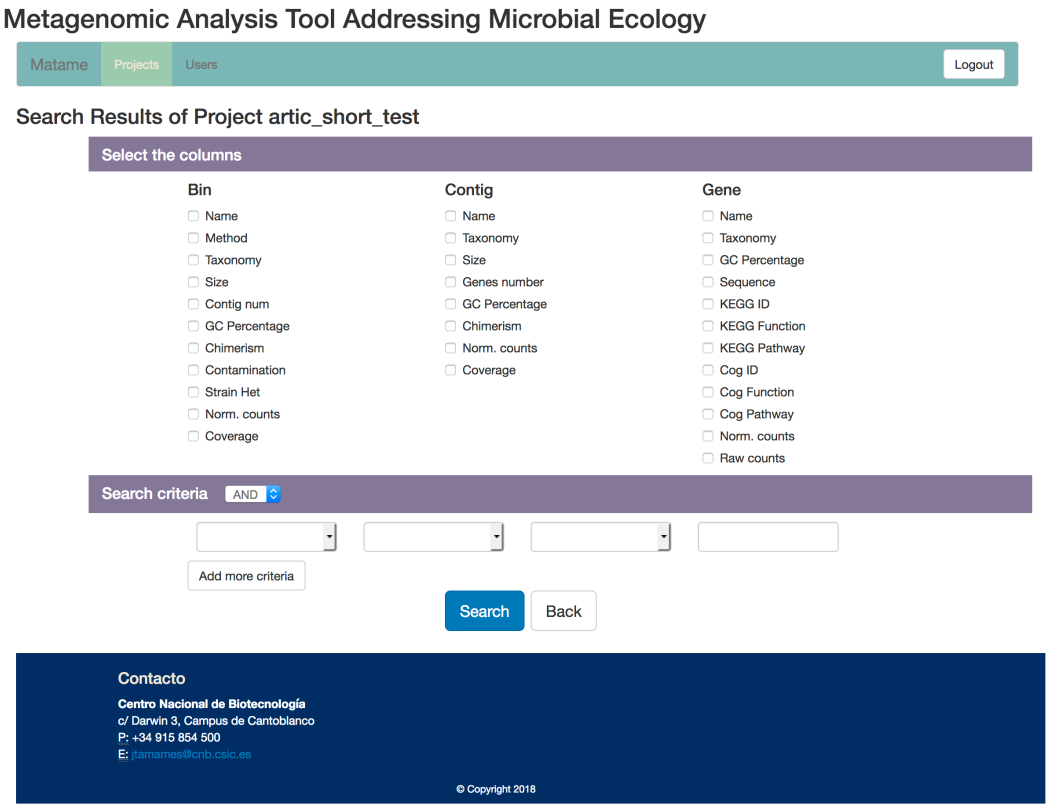

Figure 2: Snapshot of the SqueezeM database

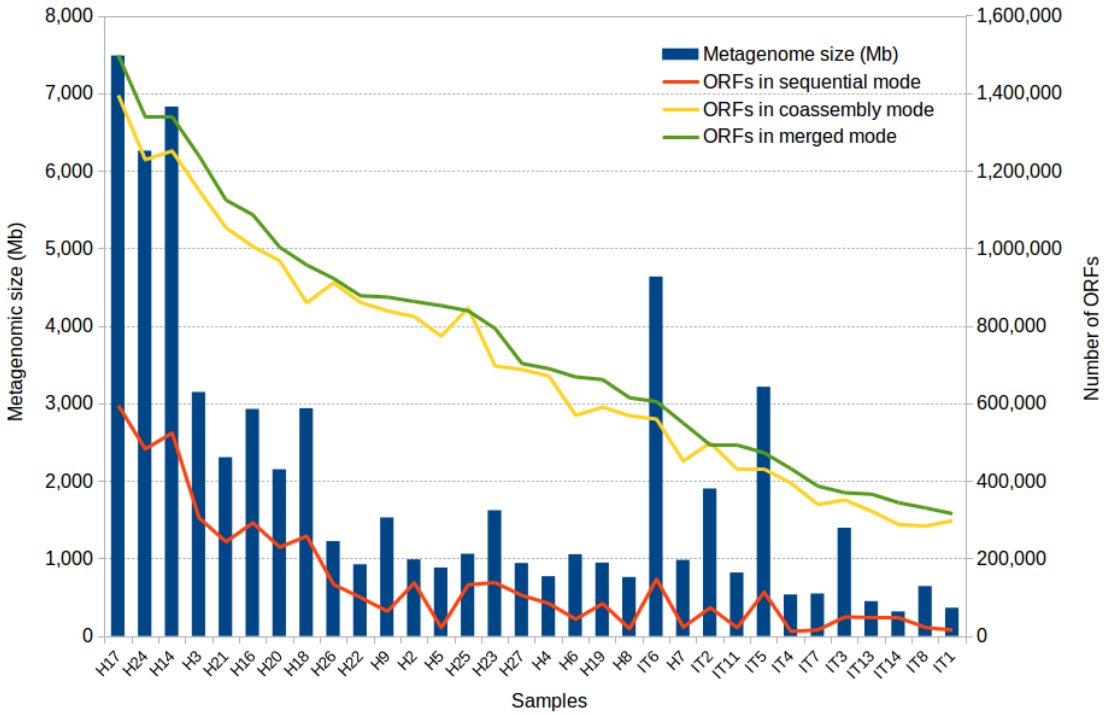

Figure 3: Size of Hadza and Italian metagenomes, and number of genes obtained using the three modes of analysis

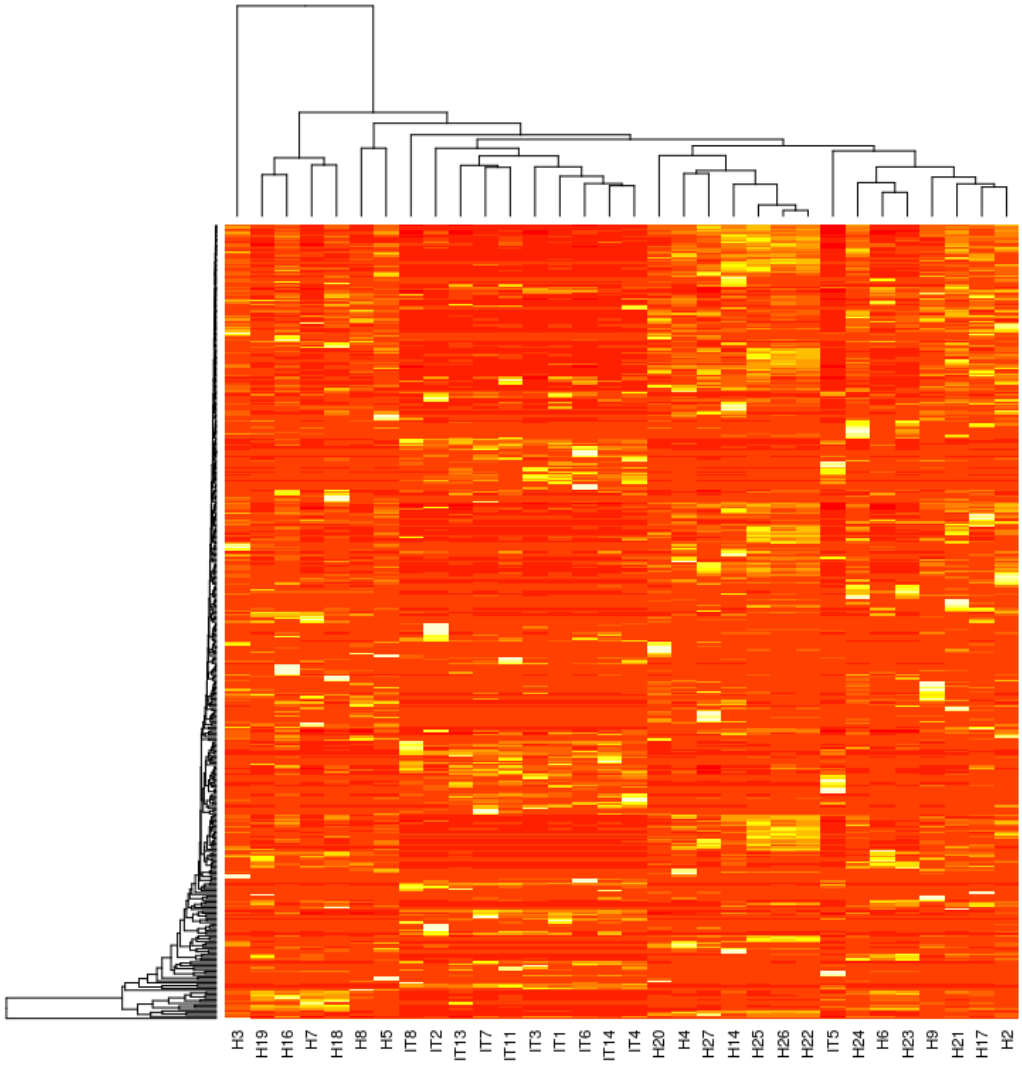

Figure 4: Abundance of bins in the diverse samples. Bins are shown in rows, while columns represent samples. Brighter (yellow) colors indicate higher abundance of that bin in the corresponding sample.