SqueezeM, a highly portable, fully automatic metagenomic analysis pipeline

Javier Tamames[1] (jtamames@cnb.csic.es) (Corresponding author)

Fernando Puente-Sánchez[1] (fpuente@ cnb.csic.es)

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

Abstract

Background
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. The bioinformatic analysis of the sequencing results is critical poses a challenge because it involves many different complex steps

Results
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. Squeeze M features several unique characteristics: Co-assembly procedure or co-assembly of unlimited number of metagenomes via merging of individual assembled metagenomes, both with read mapping for estimation of the abundances of genes in each metagenome. 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 analysis is done via mapping of cDNA reads against reference metagenomes.
The results are stored in a mySQL database, where they can be easily exported and shared, and can be inspected anywhere using a flexible web interface allowing the easy creation of complex queries.
We illustrate the potential of squeezeM by analyzing 32 gut metagenomes in a fully automatic way, allowing to retrieve several millions of genes and several hundreds of genomic bins.
One of the motivations in the development of squeezeM was producing a software capable to run in small desktop computers, thus being amenable to all users and all settings. We were also able to co-assemble two of these metagenomes and complete the full analysis in less than one day using a simple laptop computer, illustrating the capacity of SqueezeM to run without high-performance computing infrastructure. SqueezeM is therefore adequate for in-situ, real time analysis of metagenomes produced by nanopore sequencing.

Conclusions
SqueezeM is a complete system covering all steps in the analysis of metagenomes and metatranscriptomes, capable to work even in scarcity of computational resources.

Keywords

Software; Metagenomics; Binning; MinION

Background

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 already in the range of terabases), a metagenomic analysis is a complex task comprising several non-standardized steps, involving different software tools which results are often not directly compatible.
Lately, the development of the highly portable sequencers, especially these based on nanopore technologies [1], has allowed the in-situ sequencing in scenarios where the ability of obtaining quick results is paramount, for instance clinical scenarios of disease control or epidemics [2,3]. Also, metagenomic sequencing has been done in-situ, for instance in oceanographic expeditions of in the Antarctic ice [4,5], illustrating the growing capability of producing sequences right away in the sampling campaigns. This will allow the informed planning of upcoming samplings according to the results found in previous days. We foresee that this kind of application will be increasingly used in the near future. Therefore, the bioinformatic analysis should be done in a very short span of time (hours), and to be amenable to lightweight computing infrastructure.

A standard metagenomic pipeline would involve read curation, assembly, gene prediction, and functional and taxonomic annotation of the resulting genes. Several pipelines have been created for automatizing most of these analyses [6–11]. However, they differ in capacities and, most importantly, in the way they perform annotations. 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 [12]; and importantly, the sequences are often too short to produce accurate assignments [13,14]

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 and its context makes its functional and taxonomic assignment much easier and more trustable. The drawback of assembly is 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 (often below 20%). 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 [15]. Also, co-assembling related metagenomes can alleviate very much this problem, as we will discuss in the results section.

Assembly is also 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 a microbiome, since it allows to link functions to the individual organisms 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 (for instance, looking for metabolic complementations), and to advance in the understanding of the effect of ecological perturbations. The best strategy for binning is the co-assembly of related metagenomes. 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). These coherent contigs are then marked as belonging to the same bin. In this way, it is possible to retrieve tens or hundreds of genomic bins with different levels of completion, that can be used as the starting point for a more in-depth analysis of the functioning of the microbiome.

SqueezeM is a full automatic pipeline for metagenomics/metatranscriptomics, covering all steps of the analysis. A comparison of the capabilities of SqueezeM and other pipelines is shown in Table 1. One of the most notable differences is that current pipelines do not include support for coassembling and binning. At most they allow the import of the results to display binning information [7]. 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.

	MG-Rast [16]	Anvio [7]	Smash community [8]	Humann [11]	fmap [10]	MetaWrap [17]	Samsa2 [18]	squeezeM
Assembly	No	No	Yes	No	No	Yes	No	Yes
Data source	Reads or contigs	Contigs	Contigs	Reads	Reads or contigs	Contigs	Reads (RNA)	Reads
Gene prediction	Yes	Yes	Yes	No	No	No	No	Yes
Function assignment	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
RNA assignment	Yes	Yes	No	No	No	No	Yes	Yes
Taxonomic assignment	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Gene abundances	Yes	Yes	No	Yes	Yes	No	Yes	Yes
Metagomic comparison	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
Coassembly	No	No	No	No	No	Yes	No	Yes
Binning	No	Support	No	No	No	Yes	No	Yes
Local Installation	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Table 1: Features of different metagenomic analysis pipelines, in comparison to squeezeM.

Thus, squeeze M offers several unique characteristics with regard to existing pipelines:
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) Binning and bin checking, for retrieving individual genomes
4) Internal checks for the assembly and binning steps inform about the consistency of contigs and bins, allowing to spot potential chimeras.
5) Metatranscriptomic support via mapping of cDNA reads against reference metagenomes
6) Included mySQL database for storing results, where they can be easily exported and shared, and can be inspected anywhere using a web interface.

We have designed squeezeM to be able to run in scarcity of computer resources, as expected in in-situ metagenomic sequencing experiments. By setting adequately all the components of the pipeline, we demonstrate the capability of analysing completely two gut metagenomes in a desktop computer with only 16 Gb RAM. Also, the fully automatic nature of our system makes it very easy to use, not requiring any technical or bioinformatic knowledge.

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

Implementation

SqueezeM is designed to operate in multi-metagenome mode, allowing the analysis of several metagenomes in a single run. It can be run in three different modes, which are:
-Sequential mode: All metagenomes are treated individually and analysed sequentially. This mode does not include binning since the resulting contigs are not the same between metagenomes.
-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 ORF and contig abundances in each sample. Subsequent binning methods allow to obtain genomic bins.
-Merged mode: if many samples are available, it is likely that co-assembly fails because memory usage runs very high. This mode allows the co-assembly of an unlimited number of samples, using a procedure similar to the one used by Tully and colleagues in the analysis of TARA Oceans metagenomes (ref). Briefly, samples are assembled individually and the resulting contigs are re-assembled and merged in a single co-assembly. Then the analysis proceeds as in the coassembly mode.
A schematic workflow for the three modes can be seen in Figure 1.

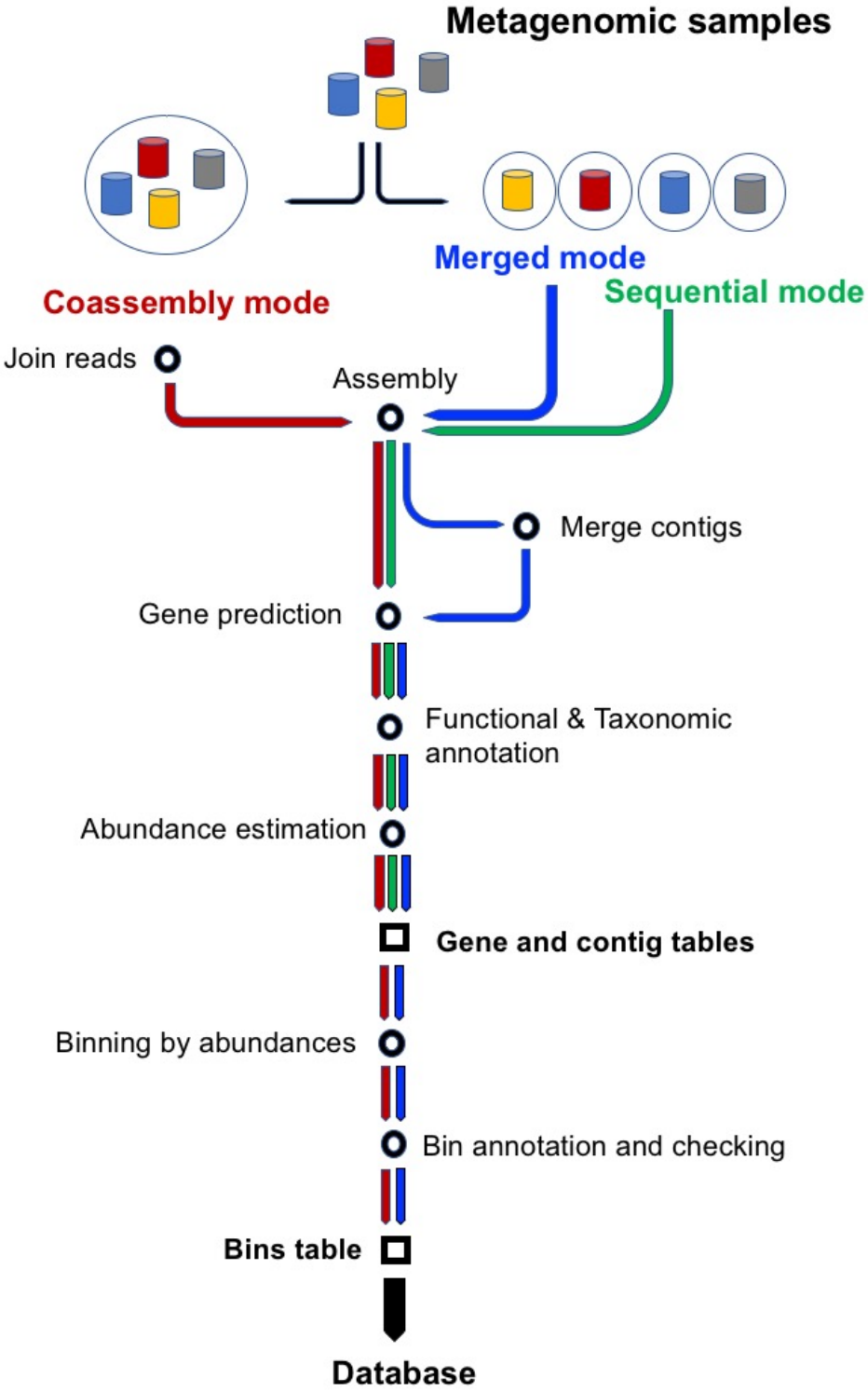

Figure 1: Workflow of the three modes of operation of SqueezeM: sequential, coassembly and merged
Starting from metagenomic samples, green, blue and red arrows indicate main steps in sequential, merged and coassembly modes. All modes create two of the three main tables of results: ORF and contig tables. Coassembly and merged modes also apply binning, and therefore they also create the bin table.

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:

• Data preparation
A squeezeM run is only requires a configuration file indicating the metagenomic samples and the location of their corresponding sequence files. The program creates the appropriate directories and prepares the data for further steps. It is important to remark that SqueezeM expects clean reads, that is, it does not perform any quality filtering on sequence reads, because the required treatments can be differ for diverse metagenomes and it is important to select the right settings accordingly.

• Assembly
When assembling big metagenomic datasets, computing resources, especially memory usage, are critical. SqueezeM uses Megahit [19] as its reference assembler, since we find it has an optimal balance between performance and memory usage. SPAdes [20] 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 [21] for removing redundant contigs and minimus2 [22] for the merging.
The resulting set of contigs is filtered by length using prinseq [23], to discard short contigs if desired. In case of adding metatranscriptomic samples, these can be flagged for no assembly but will be considered for mapping the reads against the reference metagenome.

• ORF and rRNA prediction
This step performs a ORF prediction on the resulting contigs, using the prodigal software [24], and looks for rRNAs using barrnap [25]. The program obtains ORFs and the corresponding aminoacid sequences of genes.

• Homology searching
SqueezeM uses the Diamond software [26] 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 performed: against the GenBank nr database for taxonomic assignment, against the eggNOG database [27] for COG/NOG annotation, and against the latest publicly available version of KEGG database [28] for KEGG ID annotation.

• 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 hits for each ORF against GenBank nr database. Functional assignments in COGs and KEGG IDs can be done using the classical best hit approach, or a more sensitive one considering the consistency of all hits (supplementary methods in Additional file 1)

• Taxonomic assignment of contigs and chimera check

The taxonomic assignment of individual ORFs is used to produce consensus assignments for the contigs. A contig is annotated to the taxon to which a minimal percentage of its genes belong to (Additional file 1). The required percentage can be set by the user. This procedure can accommodate missing or incorrect annotations of few genes, recent HGT events, etc.
Also a chimerism score is computed for each contig, indicating how many of the genes are discordant with the consensus (Additional file 1). Contigs with high chimerism could be the result of misassembles, and can be flagged to be excluded from subsequent analyses.

• Coverage and abundance estimation for genes and contigs
For estimating the abundance of ORFs and contigs in each of the original samples, squeezeM relies on the mapping of the original reads onto the metagenomic contigs resulting from the assembly. The software Bowtie2 [29] is used for this task, followed by Bedtools [30] for the extraction the raw number of reads and bases mapping to each ORF and contig. Custom scripts are used to compute the average coverage and normalized RPKM values informing about ORF and contig abundance.
Metatranscriptomic reads that were excluded in the assembly are used in this step to obtain an estimation of the expression level of each ORF, and a direct comparison between metagenomic and metatranscriptomic abundances.

SqueezeM stops here if it was run in sequential mode. Co-assembled metagenomes allow further binning procedures for obtaining genomes.

• 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 [31] and Metabat2 [32] are supported. SqueezeM calculates average coverage and RPKM values for the bins in the same way than above, mapping reads to the contigs belonging to the bin.

• Taxonomic assignment of bins and chimera check
As for the consensus annotation of contigs, squeezeM also generates a consensus taxonomic assignment for the bins. Equivalently to contig consensus, a bin is annotated to the taxon to which most of its contigs belong to. 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. This is a first measure of possible contaminations in the bin.

• Bin check
The goodness of the bins is estimated using the checkM software [33]. Briefly, CheckM provides indications of the completeness, contamination and strain heterogeneity of a bin by creating a profile of single-copy, conserved genes for the given taxon and evaluating how many of these genes were found (completeness), and how many were single-copy (contamination and strain heterogeinity). SqueezeM automatizes CheckM runs for each bin, using the consensus annotation for the bin as the suggested taxonomic origin.

• Merging of results

Finally, the system merges all these results and generates several tables: 1) a ORF table, with all the information regarding ORFs (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).

• 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 (ORFs, 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.

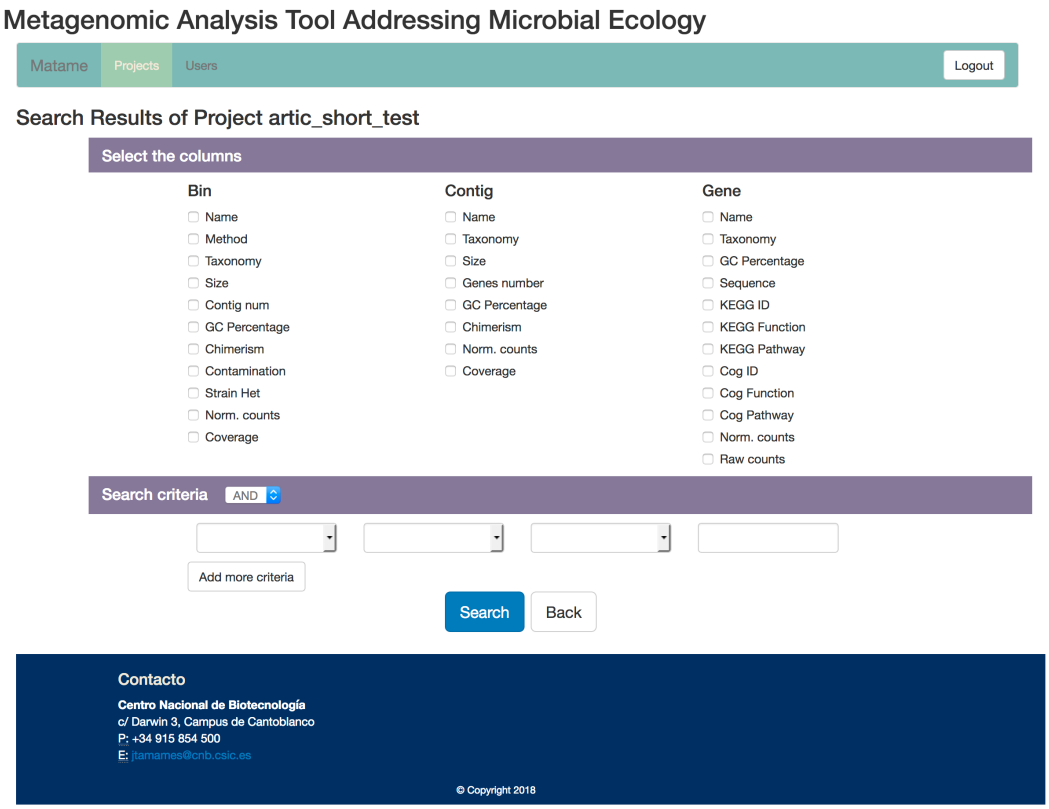

Figure 2: Snapshot of the SqueezeM user interface to its database
A flexible and intuitive system for building queries allows to interrogate the database with complex questions involving combination of data from different tables.

Results

For illustrating the use of the SqueezeM software, we analysed 32 metagenomic samples corresponding to gut microbiomes of Hadza and Italian subjects [34], using the three modes of analysis. 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. A powerful machine like this is advisable but not necessary. We were able to run analyses on merged mode using just 24 Gb RAM. 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 (they are represented by just one contig product of the co-assembly) but more than one in the individual samples (they are present in one different contig per sample). A more accurate comparison is shown in Figure 3, where a gene in the co-assembly is assumed to be present in a given 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 represented by a few reads in a sample 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.
The improvement of gene recovery for the smaller samples is also noticeable by the percentage of mapped reads. The individual assembly for small samples achieves barely 35% of read mapping to the assembled metagenome, indicating that most of the reads could not be used. The small size (and therefore low coverage) of the metagenome prevented these reads to be assembled. When co-assembling these samples with the rest, more that 85% of the reads could then be mapped to the reference metagenome, indicating that the co-assembly is able to capture most of the diversity found in these small samples.

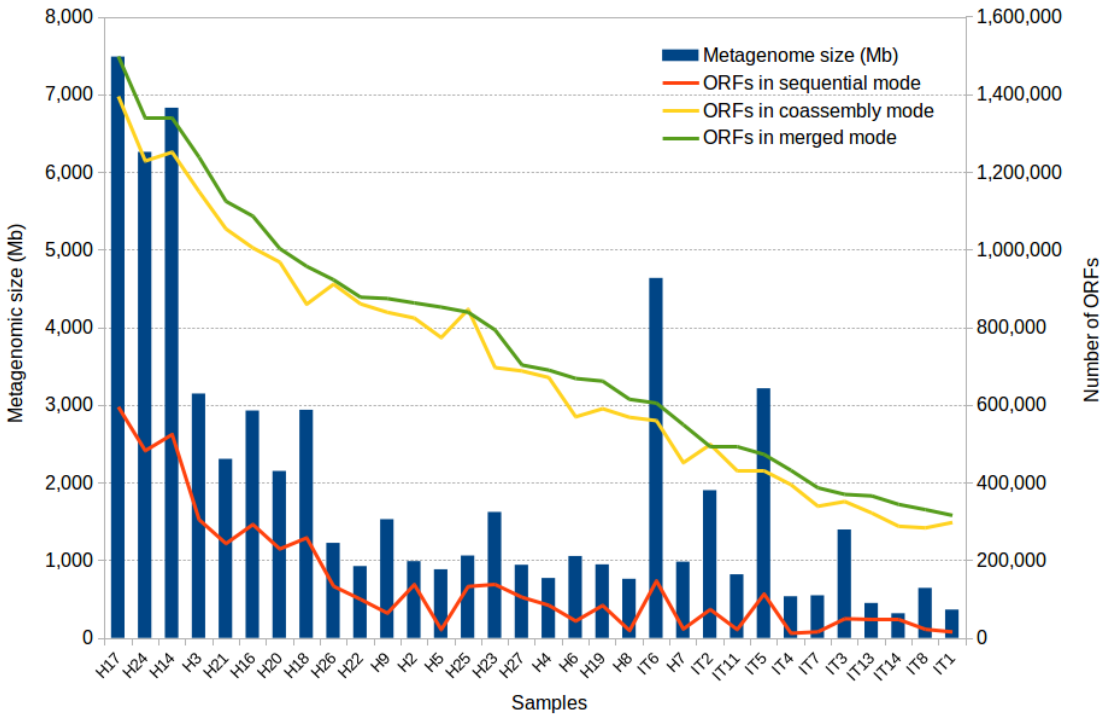

Figure 3: Results of the application of squeezeM to 32 gut metagenomes of Hadza and Italian subjects.
Size of Hadza and Italian metagenomes, and number of genes obtained using the three modes of analysis

Table 2 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 flagged when genes in the contig have been annotated to different taxa. 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 because of its lower computational burden.

	Merged mode	Co-assembly mode	Sequential mode
Number of contigs	893438	983350	2478560
N50	3900	2357	2854
Average perc of mapped reads	85.01	89.47	74.47
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
Genes with COG function	1098635 (45.7%)	982029 (44.0%)	2164980 (46.9%)
Genes with KEGG function	835498 (34.8%)	749892 (33.6%)	1683636 (36.5%)
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 2: Statistics on contigs and bins for the three modes on squeezeM on Hadza & Italian metagenomes. Binning statistics refer to MaxBin results.

The results of binning have been analysed according the values of completeness and contamination provided by checkM (Table 3). 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.

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

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 them clustering together. Bins belonging to the genera Bacteoides and Faecalibacterium are much more abundant in these individuals than in Hadza individuals. The Hadza have an increased diversity, and fall in different groups corresponding to the presence of diverse species, in accordance to the distinctions found using functional profiles [34]. The microbiota of these individuals contains genera such as Allistipes or Prevotella that are not present in the Italian metagenomes. Also, Spirochaetes from the genera Treponema are only present in Hadza subjects, where are supposedly not associated to pathogenesis. This information is directly retrieved from squeezeM results, and offers a very informative view of the genomic composition and differences between the samples.

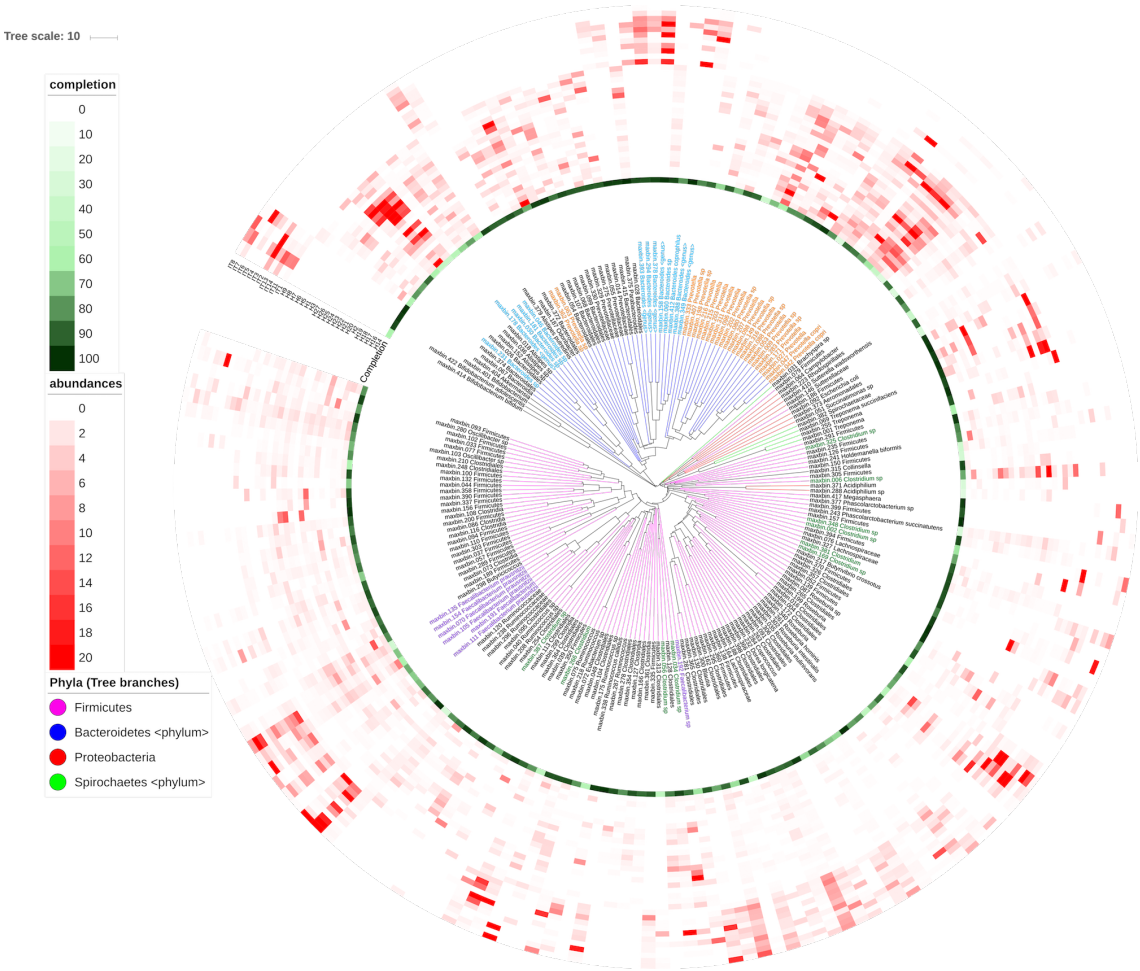

Figure 4: Abundance of bins in the diverse samples.
Red vertical strips indicate the abundance (coverage) of each bin in the Hadza (H) and Italian (IT) metagenomes (circles). The green strip indicates the completion of the bins. The tree of bins is done from a distance matrix of the amino acid identity of their orthoulogous genes, using the compareM software (https://github.com/dparks1134/CompareM). The four more abundant phyla are coloured (branches in the tree), as well as the most abundant genera (bin labels). The picture was done using the iTOL software (https://itol.embl.de)

One of the motivations for the development of squeezeM was making it capable to perform a complete metagenomic analysis on a limited computing infrastructure, like the one that can be expected in the course of in-situ metagenomic sequencing [4,5]. We developed an analysis mode carefully tailored to run with limited amounts of RAM memory. To test this capability, we were able to co-assemble two metagenomic samples from the Hadza metagenomes, composed of 30 millions of reads amounting almost 4Gb of DNA sequence. We ran the sequential mode of squeezeM using the --low-memory option in a standard desktop computer, using just 6 CPUs and 16 Gb RAM. The run completed in less than one day, generating 33.660 contigs in 38 bins and 124.065 functionally and taxonomically annotated genes. This experiment illustrates that squeezeM can be run even with scarce computational resources, not needing supercomputing infrastructure.

Conclusions

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 co-assembling 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, allowing the creation of complex queries in a very simple way.
One of the most remarkable features of this software is its capability to operate in limited computing infrastructure. We were able to analyse two gut metagenomes in a few hours using a standard desktop computer with just 16 Gb RAM. Therefore, squeezeM is apt to be used in scenarios in which computing resources are limited, such as remote locations in the course of metagenomic sampling campaigns.

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.

Availability and requirements

Project name: SqueezeM
Project home page: https://github.com/jtamames/SqueezeM
Operating systems: Linux
Programming language: Perl
License: GNU GPL
Any restrictions to use by non-academics**:** None

Acknowledgements

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

Supporting information

Additional file 1: Description of novel algorithms implemented in squeezeM.

References

1. Deamer D, Akeson M, Branton D. Three decades of nanopore sequencing. Nat Biotechnol. 2016;34:518–24.
2. Quick J, Ashton P, Calus S, Chatt C, Gossain S, Hawker J, et al. Rapid draft sequencing and real-time nanopore sequencing in a hospital outbreak of Salmonella. Genome Biol. 2015;16:114.
3. Quick J, Loman NJ, Duraffour S, Simpson JT, Severi E, Cowley L, et al. Real-time, portable genome sequencing for Ebola surveillance. Nature. 2016;530:228–32.
4. Lim YW, Cuevas DA, Silva GGZ, Aguinaldo K, Dinsdale EA, Haas AF, et al. Sequencing at sea: challenges and experiences in Ion Torrent PGM sequencing during the 2013 Southern Line Islands Research Expedition. PeerJ. 2014;2:e520.
5. Johnson SS, Zaikova E, Goerlitz DS, Bai Y, Tighe SW. Real-time DNA sequencing in the antarctic dry valleys using the Oxford nanopore sequencer. J Biomol Tech. 2017;28:2–7.
6. 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. p. 325–31.
7. Eren a. M, Esen C, Quince C, Vineis JH, Morrison HG, Sogin ML, et al. Anvi ' o : an advanced analysis and visualization platform for ' omics data. PeerJ [Internet]. 2015;3:1–29.
8. Arumugam M, Harrington ED, Foerstner KU, Raes J, Bork P. SmashCommunity: A metagenomic annotation and analysis tool. Bioinformatics. 2010;26:2977–8.
9. Li W. Analysis and comparison of very large metagenomes with fast clustering and functional annotation. BMC Bioinformatics. 2009;10:359.
10. Kim J, Kim MS, Koh AY, Xie Y, Zhan X. FMAP: Functional Mapping and Analysis Pipeline for metagenomics and metatranscriptomics studies. BMC Bioinformatics. 2016;17.
11. Abubucker S, Segata N, Goll J, Schubert AM, Izard J, Cantarel BL, et al. Metabolic reconstruction for metagenomic data and its application to the human microbiome. PLoS Comput Biol. 2012;8.
12. Pignatelli M, Aparicio G, Blanquer I, Hernández V, Moya A, Tamames J. Metagenomics reveals our incomplete knowledge of global diversity. Bioinformatics. 2008. p. 2124–5.
13. Wommack KE, Bhavsar J, Ravel J. Metagenomics: Read length matters.

Appl Environ Microbiol. 2008;74:1453–63.
14. Carr R, Borenstein E. Comparative analysis of functional metagenomic annotation and the mappability of short reads. PLoS One. 2014;9:e105776.
15. Hitch TCA, Creevey CJ. Spherical: An iterative workflow for assembling metagenomic datasets. BMC Bioinformatics. 2018;19.
16. Meyer F, Paarmann D, D'Souza M, Etal. The metagenomics RAST server—a public resource for the automatic phylo- genetic and functional analysis of metagenomes. BMC Bioinformatics. 2008;
17. Uritskiy G V, DiRuggiero J, Taylor J. MetaWRAP - a flexible pipeline for genome-resolved metagenomic data analysis. bioRxiv. 2018;
18. Westreich ST, Treiber ML, Mills DA, Korf I, Lemay DG. SAMSA2: A standalone metatranscriptome analysis pipeline. BMC Bioinformatics. 2018;
19. Li D, Liu CM, Luo R, Sadakane K, Lam TW. MEGAHIT: An ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. Bioinformatics. 2015;31:1674–6.
20. Bankevich A, Nurk S, Antipov D, Gurevich AA, Dvorkin M, Kulikov AS, et al. SPAdes: A New Genome Assembly Algorithm and Its Applications to Single-Cell Sequencing. J Comput Biol [Internet]. 2012;19:455–77.
21. Fu L, Niu B, Zhu Z, Wu S, Li W. CD-HIT: Accelerated for clustering the next-generation sequencing data. Bioinformatics. 2012;28:3150–2.
22. Sommer DD, Delcher AL, Salzberg SL, Pop M. Minimus: A fast, lightweight genome assembler. BMC Bioinformatics. 2007;8.
23. Schmieder R, Edwards R. Quality control and preprocessing of metagenomic datasets. Bioinformatics. 2011;27:863–4.
24. Hyatt D, Chen GL, LoCascio PF, Land ML, Larimer FW, Hauser LJ. Prodigal: Prokaryotic gene recognition and translation initiation site identification. BMC Bioinformatics. 2010;11.
25. Seemann T. Prokka: Rapid prokaryotic genome annotation. Bioinformatics. 2014;30:2068–9.
26. Buchfink B, Xie C, Huson DH. Fast and sensitive protein alignment using DIAMOND. Nat Methods [Internet]. 2015;12:59–60.
27. Huerta-Cepas J, Szklarczyk D, Forslund K, Cook H, Heller D, Walter MC, 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.
28. Kanehisa M, Goto S. KEGG: Kyoto Encyclopaedia of Genes and Genomes. Nucl Acids Res [Internet]. 2000;28:27–30.
29. Langmead B, Salzberg SL. Fast gapped-read alignment with Bowtie 2. Nat. Methods. 2012. p. 357–9.
30. Quinlan AR, Hall IM. BEDTools: A flexible suite of utilities for comparing genomic features. Bioinformatics. 2010;26:841–2.
31. Wu YW, Simmons BA, Singer SW. MaxBin 2.0: An automated binning algorithm to recover genomes from multiple metagenomic datasets. Bioinformatics. 2015;32:605–7.
32. Kang DD, Froula J, Egan R, Wang Z. MetaBAT, an efficient tool for accurately reconstructing single genomes from complex microbial communities. PeerJ [Internet]. 2015;3:e1165.
33. Parks DH, Imelfort M, Skennerton CT, Hugenholtz P, Tyson GW. CheckM: Assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. Genome Res. 2015;25:1043–55.

34. Rampelli S, Schnorr SL, Consolandi C, Turroni S, Severgnini M, Peano C, et al. Metagenome Sequencing of the Hadza Hunter-Gatherer Gut Microbiota. Curr Biol. 2015;25:1682–93.