VTAM: A robust pipeline for validating metabarcoding data using internal controls

Aitor González¹, Vincent Dubut², Emmanuel Corse^{3,4}, Reda Mekdad^{1,2}, Thomas Dechatre^{1,2} and Emese Megléc²

¹ Aix Marseille Univ, INSERM, TAGC, Turing Center for Living Systems, 13288 Marseille, France

² Aix Marseille Univ, Avignon Université, CNRS, IRD, IMBE, Marseille, France

³ Centre Universitaire de Mayotte, Route Nationale 3, BP 53, 97660 Dembeni, Mayotte,

France

⁴ MARBEC, CNRS, Ifremer, IRD, University of Montpellier, Montpellier, France

Corresponding author: Aitor González (aitor.gonzalez@univ-amu.fr) and Emese Megléc (emese.meglecz@imbe.fr)

Running title: VTAM metabarcoding pipeline

Abstract

1. Metabarcoding studies should be carefully designed to minimize false positives and false negative occurrences. The use of internal controls, replicates, and several overlapping markers is expected to improve the bioinformatics data analysis.
2. VTAM is a tool to perform all steps of data curation from raw fastq data to taxonomically assigned ASV (Amplicon Sequence Variant or simply variant) table. It addresses all known technical error types and includes other features rarely present in existing pipelines for validating metabarcoding data: Filtering parameters are obtained from internal control samples; cross-sample contamination and tag-jump are controlled; technical replicates are used to ensure repeatability; it handles data obtained from several overlapping markers.
3. Two datasets were analysed by VTAM and the results were compared to those obtained with a pipeline based on DADA2. The false positive occurrences in samples were considerably higher when curated by DADA2, which is likely due to the lack of control for tag-jump and cross-sample contamination.
4. VTAM is a robust tool to validate metabarcoding data and improve traceability, reproducibility, and comparability between runs and datasets.

Keywords: metabarcoding, mock sample, negative control, replicates, taxonomic assignation, false positives, false negatives

1 Introduction

Metabarcoding has become a powerful approach to study biodiversity from environmental samples (including gut content or faecal samples). Metabarcoding, however, is prone to some pitfalls, and consequently, every metabarcoding study should be designed in a from-benchtop-to-desktop way (from sampling to data analysis) to minimize the bias of each step on the outcome (Alberdi, Aizpurua, Gilbert, & Bohmann, 2018; Cristescu & Hebert, 2018; Zinger et al., 2019). Several papers have called for good practice in study design, data production and analyses to ensure repeatability and comparability between studies. Notably, the importance of mock community samples, negative controls, and replicates is frequently highlighted (Alberdi et al., 2018; Bakker, 2018; Cristescu & Hebert, 2018;

O'Rourke, Bokulich, Jusino, MacManes, & Foster, 2020). However, their use in bioinformatics pipelines is often limited to the verification of expectations. In this study, we present the bioinformatics pipeline, VTAM (Validation and Taxonomic Assignment of Metabarcoding data) that effectively integrates negative controls, mock communities and technical replicates to control experimental fluctuations (e.g. sequencing depth, PCR stochasticity) and validate metabarcoding data. A recent study on the effect of different steps of data curation on spatial partitioning of biodiversity listed the following potential problems: Sequencing and PCR errors, presence of highly spurious sequences, chimeras, internal or external contamination and dysfunctional PCRs (Calderón-Sanou, Münkemüller, Boyer, Zinger, & Thuiller, 2020). They showed that exhaustive curation and ensuring repeatability by technical replicates are necessary, especially for biodiversity measurements. Ideally, a metabarcoding workflow should address all of these technical errors. Existing tools, however, are either highly flexible but too complex or they do not include the curation of all potential biases (Mahé, Rognes, Quince, de Vargas, & Dunthorn, 2014; Boyer et al., 2016; Callahan et al., 2016; Edgar, 2016b; Rognes, Flouri, Nichols, Quince, & Mahé, 2016; Bolyen et al., 2019). The filtering steps of VTAM aim to address these points and include several additional features that are unique or rarely found in existing pipelines: (i) the use of internal controls and (ii) replicates to optimize filtering parameter values; (iii) the integration of multiple overlapping markers and (iv) filtration to address cross-sample contamination, including tag-jumps. Finally, VTAM is a variant-based filtering pipeline (such as other denoising methods: Callahan et al., 2016; Edgar, 2016b) that deals with amplicon sequence variants (ASVs).

2 Features

2.1 Implementation

VTAM is based on the method described in Corse et al. 2017. It is a command-line application that runs on Linux, MacOS or Windows Subsystem for Linux (WSL). VTAM is implemented in Python3, using a Conda environment to ensure repeatability and easy installation of VTAM and these third-party applications: WopMars (<https://wopmars.readthedocs.io>), NCBI BLAST, Vsearch (Rognes et al., 2016), Cutadapt (Martin, 2011). Data is stored in an SQLite database that ensures traceability.

86

87 2.2 Workflow

88 Table 1 summarizes the different commands and steps of VTAM, their purpose and
89 the related error types.

90 2.2.1 Pre-processing (optional)

91 An example of the data structure is illustrated in Fig. 1.

92 Paired-end FASTQ files are merged, reads are trimmed and demultiplexed
93 according to forward and reverse tag combinations.

94 2.2.2 Filtering

95 Demultiplexed reads are dereplicated and ASVs are stored in an SQLite database.

96 All occurrences are characterized by their read count.

97 *FilterLFN*: eliminates occurrences likely due to Low Frequency Noise. Occurrences
98 are filtered out if they have low read counts (i) in absolute terms (N_{ijk} is small,
99 where N_{ijk} is the read count of variant i in sample j and replicate k), (ii) compared
100 to the total number of reads of the sample-replicate (N_{ijk}/N_{jk}) or (iii) compared to
101 the total number of reads of the variant (N_{ijk}/N_i).

102 *FilterMinReplicateNumber*: Occurrences are retained only if the ASV is present in
103 at least a user-defined number of replicates.

104 *FilterPCRError*: ASVs with one difference from another ASV of the same sample
105 are filtered out if the proportion of their read counts is below a user-defined
106 threshold value.

107 *FilterChimera* runs the uchime3_denovo chimera filtering implemented in *vsearch*.

108 *FilterRenkonen* removes whole replicates that are too different compared to other
109 replicates in the same sample.

110 *FilterIndel* and *FilterCodonStop* are intended to detect pseudogenes and should
111 only be used for coding markers. *FilterIndel* eliminates all variants, with aberrant
112 length, where the modulo three of the length is different from the majority.

113 *FilterCodonStop* eliminates all variants that have codon STOP in all reading frames
114 of the direct strand.

115 The output of the filters is an ASV table with validated variants in lines, samples in
116 columns and the sum of read counts over replicates in the cells.

117 2.2.3 Taxonomic assignation

118 Taxonomic assignation is based on the Lowest Taxonomic Group method described
119 in detail in Supporting Information 1. The taxonomic reference database has a
120 BLAST format with taxonomic identifiers so that custom databases or the complete
121 NCBI nucleotide database can be used by VTAM. A custom taxonomic reference

database of COI sequences mined from NCBI nucleotide and BOLD
(<https://www.boldsystems.org/>) databases is available with the program.

2.2.4 Parameter optimization

Users should first identify expected and unexpected occurrences based on the first filtration with default parameters. The optimization step will guide users to choose parameter values that maximize the number of expected occurrences in the dataset and minimize the number of unexpected occurrences (false positives). Parameters are optimized for the three LFN filters and the FilterPCRError. Optimized parameters can then be used to repeat the filtering steps.

2.2.5 Pool runs/markers

A run is FASTQ data from a sequencing run and a marker is a region of a locus amplified by a primer pair. The pool command produces an ASV table with any number of run-marker combinations. When more than one overlapping marker is used, ASVs identical to their overlapping parts are pooled to the same line.

3 Benchmarking

VTAM was tested with two published metabarcoding datasets: a fish dataset obtained from fish faecal samples (Corse et al., 2017), and a bat dataset obtained from bat guano samples (Galan et al., 2018). Both datasets included negative controls, mock samples and three PCR replicates. A fragment of the COI gene was amplified using two overlapping markers in the fish dataset, and one in the bat dataset (See details in the original studies).

Both datasets were analysed by VTAM. The fish dataset was analysed separately for the two markers and the results of both markers were pooled together.

Both datasets were also analysed with the DADA2 denoising algorithm (Callahan et al., 2016), one of the most widely used methods for metabarcoding data curation. The output of DADA2 was filtered by LULU (Frøslev et al., 2017) to further eliminate probable false positive occurrences. Then the three replicates of each sample were pooled (as in VTAM), only accepting the occurrence if it was present in at least two replicates (Supporting information 2).

We compared the α -diversity and β -diversity obtained for the environmental samples to address the effect of the curation pipelines on diversity estimations. α -diversity was estimated using both ASV richness and cluster richness (clusters aggregate ASVs with <3% divergence), and β -diversity was summarized using the Bray-Curtis pairwise dissimilarity index. (Supporting information 3).

In the fish dataset, all expected variants in the mock samples were validated by both pipelines. However, in the bat dataset, two expected variants had very low

158 read abundance (2-18 reads/replicate), which were in the range of the number of
159 reads in the negative controls (ten out of the 19 negative controls had at least one
160 read count greater than 18). Therefore, we ignored these two expected variants in
161 the Bulk France mock sample, and we optimized the VTAM parameters to retain all
162 other expected occurrences.

163 After filtering with VTAM, the number of false positives in the mock samples was
164 markedly lower than with DADA2 (Table 2). Similarly, ASV and cluster richness
165 were on average two times lower with VTAM than with DADA2 in environmental
166 samples (Fig. 2A and B). In contrast, dissimilarities between samples were higher
167 with VTAM (Fig. 2D). In both pipelines, most clusters contained a single ASV
168 (Supporting information 3; Fig. 2C).

169 4 Discussion

170 Metabarcoding is known to be prone to two types of errors: false negatives and
171 false positives. Based on controls (negative and mock samples), VTAM aims to find
172 a compromise between these two error types by minimizing false positive
173 occurrences while retaining expected variants in mock samples to avoid false
174 negatives. Therefore, the mock samples should contain both well and weakly
175 amplified taxa, where the abundance, i.e. the number of reads, of weakly amplified
176 taxa is marginally higher than those found in negative samples. This should ensure
177 finding filtering parameter values that simultaneously minimize false positives and
178 false negatives. Additionally, in large-scale studies with more than one sequencing
179 run, the use of identical mock samples in all runs can ensure comparability among
180 runs if they consistently yield the same results.

181 The use of technical replicates is another important tool to limit false positives and
182 false negatives (Alberdi et al. 2018, Corse et al. 2017). False positives can be
183 strongly reduced by only accepting variants in a sample if they are present in at
184 least a certain number of replicates. This strategy is strongly advised to reduce
185 experimental stochasticity and validate ASV occurrences. Furthermore, removing
186 replicates with radically different compositions (Renkonen filter) further reduces
187 the effect of experimental stochasticity (De Barba et al., 2014). Additionally, false
188 negatives can be further reduced by amplifying several markers (Corse et al.,
189 2019). If the different markers overlap, VTAM can pool sequences that are
190 identical in their overlapping regions. This integrates the results of different
191 markers unambiguously.

192 While false positive occurrences due to sequencing and PCR errors are generally
193 well detected by denoising pipelines such as DADA2, tag-jump and cross-sample

contamination are rarely taken into account (but see Boyer et al., 2016; Edgar, 2016a). However, failing to filter out these artefacts is likely to inflate false positive occurrences and artificially increase inter-sample similarities. In fact, the DADA2 based pipeline produced ASV and cluster richness per sample that was on average twice as high as with VTAM and even higher for some samples (Fig. 2 A, B). On the other hand, dissimilarities between samples were lower after DADA2 filtration. Additionally, the near 1:1 correlation between ASV and cluster richness in both pipelines indicated that most clusters contained just one ASV per sample. This supports the notion that diversity inflation in DADA2 resulted from unfiltered tag-jump contaminations rather than PCR or sequencing errors as this would have produced more ASVs that belong to the same cluster. Our VTAM pipeline, therefore, appears more appropriate for comparing the diversity between samples and for investigating the biological responses to environmental change.

5 Conclusions

The VTAM metabarcoding pipeline aims to address known technical errors during data analysis (Table 1) to validate metabarcoding data. It is a complete pipeline from raw FASTQ data to curated ASV tables with taxonomic assignments. The implementation of VTAM provides several advantages such as using a Conda environment to facilitate the installation, data storage in SQLite database for traceability and the possibility to run one or several sequencing run-marker combinations using the same command. VTAM includes features rarely considered in most metabarcoding pipelines, and we believe it provides a useful tool for the analysis and validation of metabarcoding data for conducting robust analyses of biodiversity.

Acknowledgements

We thank Diane Zarzoso-Lacoste and Samanta Ortuno Miguel for valuable comments on the use of VTAM, Luc Giffon and Lionel Spinelli for the development of Wopmars and Kurt Villsen for English editing. Centre de Calcul Intensif d'Aix-Marseille is acknowledged for granting access to its high performance computing resources. This work is a contribution to the European project SEAMoBB, funded by ERA-Net Mar-TERA and managed by ANR (number ANR_17_MART-0001_01).

Authors' contributions

EM, EC, VD conceived the ideas and designed the methodology. EM and AG conceived the software architecture and tested the VTAM. AG, TD and RM developed the VTAM software; AG contributed to the WopMars software

229 development. EM, AG, VD and EC wrote the manuscript. All authors contributed
230 critically to the draft and approved the final version of the manuscript.
231

232 Data Availability

233 VTAM is available at <https://github.com/aitgon/vtam>. A detailed user manual is
234 found at <https://vtam.readthedocs.io>.
235 Empirical data used in this paper are available from the Dryad Digital Repository
236 <https://datadryad.org/stash/dataset/doi:10.5061/dryad.f40v5> and
237 <https://datadryad.org/stash/dataset/doi:10.5061/dryad.kv02g>.

238 References

- 239 Alberdi, A., Aizpurua, O., Gilbert, M. T. P., & Bohmann, K. (2018). Scrutinizing
240 key steps for reliable metabarcoding of environmental samples. *Methods in Ecology*
241 *and Evolution*, 9(1), 134–147. doi:10.1111/2041-210X.12849
- 242 Bakker, M. G. (2018). A fungal mock community control for amplicon sequencing
243 experiments. *Molecular Ecology Resources*, 18(3), 541–556. doi:10.1111/1755-
244 0998.12760
- 245 Bolyen, E., Rideout, J. R., Dillon, M. R., Bokulich, N. A., Abnet, C. C., Al-
246 Ghalith, G. A., ... Caporaso, J. G. (2019). Reproducible, interactive, scalable and
247 extensible microbiome data science using QIIME 2. *Nature Biotechnology*, 37(8),
248 852–857. doi:10.1038/s41587-019-0209-9
- 249 Boyer, F., Mercier, C., Bonin, A., Bras, Y. L., Taberlet, P., & Coissac, E. (2016).
250 obitools: a unix-inspired software package for DNA metabarcoding. *Molecular*
251 *Ecology Resources*, 16(1), 176–182. doi:10.1111/1755-0998.12428
- 252 Calderón-Sanou, I., Münkemüller, T., Boyer, F., Zinger, L., & Thuiller, W. (2020).
253 From environmental DNA sequences to ecological conclusions: How strong is the
254 influence of methodological choices? *Journal of Biogeography*, 47(1), 193–206.
255 doi:10.1111/jbi.13681
- 256 Callahan, B. J., McMurdie, P. J., Rosen, M. J., Han, A. W., Johnson, A. J. A., &
257 Holmes, S. P. (2016). DADA2: High-resolution sample inference from Illumina
258 amplicon data. *Nature Methods*, 13(7), 581–583. doi:10.1038/nmeth.3869
- 259 Corse, E., Megléc, E., Archambaud, G., Ardisson, M., Martin, J.-F., Tougaard, C.,
260 ... Dubut, V. (2017). A from-benchtop-to-desktop workflow for validating HTS data
261 and for taxonomic identification in diet metabarcoding studies. *Molecular Ecology*
262 *Resources*, 17(6), e146–e159. doi:10.1111/1755-0998.12703
- 263 Corse, E., Tougaard, C., Archambaud-Suard, G., Agnès, J.-F., Mandeng, F. D. M.,
264 Bilong, C. F. B., ... Dubut, V. (2019). One-locus-several-primers: A strategy to
265 improve the taxonomic and haplotypic coverage in diet metabarcoding studies.
266 *Ecology and Evolution*, 9(8), 4603–4620. doi:10.1002/ece3.5063
- 267 Cristescu, M. E., & Hebert, P. D. N. (2018). Uses and Misuses of Environmental
268 DNA in Biodiversity Science and Conservation. *Annual Review of Ecology*,

- 269 *Evolution, and Systematics*, 49(1), 209–230. doi:10.1146/annurev-ecolsys-110617-
270 062306
- 271 De Barba, M., Miquel, C., Boyer, F., Mercier, C., Rioux, D., Coissac, E., &
272 Taberlet, P. (2014). DNA metabarcoding multiplexing and validation of data
273 accuracy for diet assessment: application to omnivorous diet. *Molecular Ecology*
274 *Resources*, 14(2), 306–323. doi:10.1111/1755-0998.12188
- 275 Edgar, R. C. (2016a). UNCROSS: Filtering of high-frequency cross-talk in 16S
276 amplicon reads. *BioRxiv*, 088666. doi:10.1101/088666
- 277 Edgar, R. C. (2016b). UNOISE2: improved error-correction for Illumina 16S and
278 ITS amplicon sequencing. *BioRxiv*, 081257. doi:10.1101/081257
- 279 Frøslev, T. G., Kjølner, R., Bruun, H. H., Ejrnæs, R., Brunbjerg, A. K., Pietroni, C.,
280 & Hansen, A. J. (2017). Algorithm for post-clustering curation of DNA amplicon
281 data yields reliable biodiversity estimates. *Nature Communications*, 8(1), 1–11.
282 doi:10.1038/s41467-017-01312-x
- 283 Galan, M., Pons, J.-B., Tournayre, O., Pierre, É., Leuchtman, M., Pontier, D., &
284 Charbonnel, N. (2018). Metabarcoding for the parallel identification of several
285 hundred predators and their prey: Application to bat species diet analysis.
286 *Molecular Ecology Resources*, 18(3), 474–489. doi:10.1111/1755-0998.12749
- 287 Mahé, F., Rognes, T., Quince, C., de Vargas, C., & Dunthorn, M. (2014). Swarm:
288 robust and fast clustering method for amplicon-based studies. *PeerJ*, 2, e593.
289 doi:10.7717/peerj.593
- 290 Martin, M. (2011). Cutadapt removes adapter sequences from high-throughput
291 sequencing reads. *EMBnet.Journal*, 17(1), 10–12. doi:10.14806/ej.17.1.200
- 292 O’Rourke, D. R., Bokulich, N. A., Jusino, M. A., MacManes, M. D., & Foster, J. T.
293 (2020). A total crapshoot? Evaluating bioinformatic decisions in animal diet
294 metabarcoding analyses. *Ecology and Evolution*, 10(18), 9721–9739.
295 doi:10.1002/ece3.6594
- 296 Rognes, T., Flouri, T., Nichols, B., Quince, C., & Mahé, F. (2016). VSEARCH: a
297 versatile open source tool for metagenomics. *PeerJ*, 4, e2584.
298 doi:10.7717/peerj.2584
- 299 Zinger, L., Bonin, A., Alsos, I. G., Bálint, M., Bik, H., Boyer, F., ... Taberlet, P.
300 (2019). DNA metabarcoding—Need for robust experimental designs to draw sound
301 ecological conclusions. *Molecular Ecology*, 28(8), 1857–1862.
302 doi:10.1111/mec.15060

303

304

Figures and tables

Figure 1. An example of a data structure with one run, two markers and three replicates for each sample. S1-R1: Replicate1 of Sample1. Replicates are not essential but strongly recommended. Samples should include at least one mock sample and one negative control.

Figure 2. Diversity estimates from the fish and bat datasets, based on the VTAM and DADA2-based pipelines. A) ASV richness per sample B) cluster richness per sample C) The correlation between ASV and cluster richness. P -value indicates a significant slope difference between the two pipelines. D) β -diversity was estimated using the Bray-Curtis dissimilarity index calculated for each pairwise sample comparison. Solid lines indicate linear regression lines, hatched lines are the 1:1 reference lines.

Table 1. List of VTAM commands and their roles.

VTAM command	VTAM step (Name in Corse et al. 2017)	Role	Error Type
merge		Merges paired-end reads and quality filtering	Sequencing errors
sortreads		Assigns reads to samples	Sequencing errors
filter	Dereplicate	Dereplicates	
filter	Delete singletons	Deletes singletons	Sequencing errors, highly spurious sequences
filter	LFN_variant filter (LFNtag)	Deletes low frequency errors	Tug jump, inter sample contamination
filter	LFN_read_count filter (LFNneg)	Deletes low frequency errors	Sequencing error, light contamination
filter	LFN_sample_replicate filter (LFNpos)	Deletes low frequency errors	Sequencing error, light contamination
filter	FilterMinReplicateNumber	Ensures consistency between replicates	PCR heterogeneity
filter	FilterPCRError (Obliclean)	Eliminates PCR errors (even if frequent)	PCR errors
filter	FilterChimera	Eliminates chimeras	Chimeras
filter	FilterRenkonen	Eliminates aberrant replicates	Dysfunctional PCRs
filter	FilterIndel (Pseudogene filter)	Eliminates pseudogenes	Pseudogenes, spurious sequences
filter	FilterCondonStop (Pseudogene filter)	Eliminates pseudogenes	Pseudogenes, spurious sequences
taxassign	(LTG)	Assigns variants to taxa	Highly spurious sequences
optimize	OptimizeLFNsampleReplicate	Finds the optimal parameter for the LFN-sample-replicate filter	
optimize	OptimizePCRError	Finds the optimal parameter for FilterPCRError	
optimize	OptimizeLFNreadCountAndLFN-variant	Finds the optimal value for LFN-read-count and LFN-variant filters	
pool		Pools the results from different runs/markers	

329 **Table 2.** Number of false positive occurrences compared to the total number
 330 of occurrences. In negative control and mock samples, the count of false
 331 positives is precise, since the sample composition is known.

	VTAM Fish	DADA Fish	VTAM Bat	DADA Bat
Negative controls	0/0 (0%)	32/32 (100%)	2/2 (100%)	19/19 (100%)
Mock samples	5/17 (29%)	37/49 (75%)	22/61 (36%)	73/114 (65%)

332

333

334 [Supporting Information](#)

335

336 **SuppInfo1.pdf**

337 Description of the taxonomic assignation and its custom database.

338 **SuppInfo2.pdf**

339 Commands, user input files, and the final ASV tables produced by VTAM

340 and the DADA based pipeline for the fish and the bat datasets.

341

342 **SuppInfo3.pdf**

343 Diversity estimation protocol