Molecular Ecology Resources
October 2021, Volume 21, Issue 7, Pages 2565-2579
https://doi.org/10.1111/1755-0998.13430
https://archimer.ifremer.fr/doc/00696/80761/

Benchmarking bioinformatic tools for fast and accurate eDNA metabarcoding species identification

Mathon Laetitia [1, 2, *], Valentini Alice [2], Guérin Pierre-edouard [1], Normandeau Eric [3], Noel Cyril [4], Lionnet Clément [5], Boulanger Emilie [1, 6], Thuillier Wilfried [5], Bernatchez Louis [3], Mouillot David [6, 7], Dejean Tony [2], Manel Stéphanie [1]

[1] CEFE, Univ. Montpellier, CNRS, EPHE-PSL University, IRD, Univ Paul Valéry Montpellier 3 Montpellier ,France
[2] SPYGEN, 17 rue du Lac Saint-André, Savoie Technolac 73370 Le Bourget du Lac, France
[3] Université Laval IBIS (Institut de Biologie Intégrative et des Systèmes) 1030 av. de la Médecine Québec QC G1V 0A6 ,Canada
[4] IFREMER - IRSI - Service de Bioinformatique (SeBiMER) 29280 Plouzané ,France
[5] Univ. Grenoble Alpes, Univ. Savoie Mont Blanc, CNRS, LECA, Laboratoire d'Écologie Alpine F- 38000 Grenoble ,France
[6] MARBEC, Univ. Montpellier,CNRS, IRD, Ifremer Montpellier ,France
[7] Institut Universitaire de France IUF Paris 75231, France

* Corresponding author : Laetitia Mathon, email address : laetitia.mathon@gmail.com

Abstract :

1. Bioinformatic analysis of eDNA metabarcoding data is crucial toward rigorously assessing biodiversity. Many programs are now available for each step of the required analyses, but their relative abilities at providing fast and accurate species lists have seldom been evaluated.

2. We used simulated mock communities and real fish eDNA metabarcoding data to evaluate the performance of 13 bioinformatic programs and pipelines to retrieve fish occurrence and read abundance using the 12S mt rRNA gene marker. We used four indices to compare the outputs of each program with the simulated samples: sensitivity, F-measure, root-mean-square error (RMSE) on read relative abundances, and execution time.

3. We found marked differences among programs only for the taxonomic assignment step, both in terms of sensitivity, F-measure and RMSE. Running time was highly different between programs for each step. The fastest programs with best indices for each step were assembled into a pipeline. We compare this pipeline to pipelines constructed from existing toolboxes (OBITools, Barque, and QIIME 2). Our pipeline and Barque obtained the best performance for all indices and appear to be better alternatives to highly used pipelines for analyzing fish eDNA metabarcoding data with a complete reference database. Real eDNA metabarcoding data also indicated differences for taxonomic assignment and execution time only.

4. This study reveals major differences between programs during the taxonomic assignment step. The choice of algorithm for the taxonomic assignment can have a significant impact on diversity estimates and should be made according to the objectives of the study.

Keywords : benchmark, bioinformatics, eDNA, metabarcoding, sensitivity, species identification

1. Introduction

Environmental DNA (eDNA) metabarcoding is a promising approach to identify species within communities and can be used to evaluate biodiversity through a variety of estimators (Boulanger et al., 2021; Deiner et al., 2020; Pawlowski et al., 2018). The approach is based on the collection of environmental samples (e.g. soil, air or water) that contain the target organisms' DNA. After DNA extraction, DNA amplification with primers designed for a specific taxonomic group is performed and submitted to high-throughput sequencing (Deiner et al., 2017; Taberlet et al., 2018). The resulting sequencing data typically contains millions of amplicon DNA fragments. Bioinformatic programs are then used to i) clean the data, ii) associate fragments to samples when amplicons are pooled in a single library and iii) produce either a matrix of read counts per species occurring within each sample using a reference database or a matrix of operational taxonomic units (OTUs) occurring within each sample. With decent completeness of the genetic reference database, eDNA metabarcoding can provide accurate representation of the taxonomic composition within samples (Djurhuus et al., 2020; Marques et al., 2020; Minamoto et al., 2012). Nevertheless, many biases can reduce the performance of such approaches which need to be controlled for, such as PCR and sequencing errors, gaps in reference databases, different species with identical sequences in the amplified region, etc. (Kwok & Higuchi, 1989; Schnell, Bohmann, & Gilbert, 2015; Zinger et al., 2019). Several of these biases mitigated with posterior bioinformatics analyses by implementing appropriate filters. Yet comparative quantitative analyses are still lacking on the different key steps of bioinformatics pipelines.

A literature search with the keywords "environmental DNA", "metabarcoding", "community", "bioinformatics analysis" (Supporting Information Method S1) identified six steps from raw sequence fragments to final identifications: paired-end reads merging, demultiplexing, dereplication, quality filtering, removal and correction of PCR/sequencing errors and taxonomic assignment (Fig 1). The order of those steps can vary depending on the pipeline being used. Some of these steps can have a strong impact on the resulting taxonomic composition and consequently, on biodiversity estimation (Bonder et al., 2012; Calderón-Sanou et al, 2020). Therefore, choosing the most appropriate bioinformatics program producing accurate, fast and sensitive taxa identifications is crucial (Pauvert et al., 2019). Until now, no consensus among existing bioinformatics programs and pipelines has emerged to choose the most appropriate for analyzing eDNA metabarcoding data (Bazinet & Cummings, 2012; Gardner et al., 2019; Lindgreen et al., 2016; Peabody et al., 2015; Prodan et al., 2020; Sczyrba et al., 2017; Siegwald et al., 2017). In particular, there is no standard or recommendation related to existing programs, based on quantitative comparisons, in the context of aquatic eDNA data and particularly so for Teleostean fishes.

With more than 32,000 species, Teleostean fishes are the largest group of vertebrates (www.fishbase.org). Worldwide, a growing number of fish species and populations are threatened and decreasing in size due to overfishing as well as habitat degradation (Yan et al., 2021). With more than 60% of the publications on eDNA dealing with vertebrate monitoring, fishes represent the most studied group using eDNA approaches (Tsuji et al., 2019). As a result, they represent a relevant candidate taxonomic group to compare eDNA metabarcoding programs and pipelines.

eDNA metabarcoding based on water samples was first applied to monitor fish species both in marine (Thomsen et al., 2012) and freshwater environments (Robson et al., 2016). Currently, there is increasing interest in this technique for characterizing fish diversity (Berger et al., 2020; Jerde, Wilson, & Dressler, 2019; Juhel et al., 2020; McElroy et al., 2020; Sigsgaard et al., 2017) in particular when classical methods are too invasive or do not perform well, as is the case for rare species or those that inhabit the deep sea. Many primer pairs have been developed to amplify different mitochondrial DNA fragments of fish DNA. The primer pair used in this study (teleo 12S mt rRNA; Valentini et al., 2016) is one of the most frequently used and has been proven effective in detecting rare biodiversity and discriminating species in European fluvial ecosystems (Civade et al., 2016; Collins et al., 2019; Pont et al., 2019). Since teleo 12S mt rRNA is a widely used primer set, it is a good candidate to explore the efficiency of different bioinformatics programs.

In this context, the goals of this study were to: i) explore common bioinformatic tools including one-step programs used in the different steps of metabarcoding data analysis and integrated pipelines, ii) assess the ability of these programs to retrieve accurately and rapidly species composition (occurrences and abundances) of representative fish communities, iii), assemble the best-performing programs for each step into a custom assembled pipeline, and compare its performance to three other pipelines designed for metabarcoding analysis (Barque, OBITools and QIIME2-based) using both simulated and real data.

2. Material and methods

2.1. Simulated fish communities

We simulated 29 different species assemblages, hereafter designed as “samples”, using the program Grinder (Angly et al., 2012). In these samples, the presence of 18 to 51 fish species were included (among a variety of cartilaginous and bony fishes: Actinopteri, Chondrichthyes, Cladistia, Cyclostomata and Sarcopterygii). Each sample was composed of random species from various classes, orders and families. Some samples contained several species belonging to the same genus (samples 3, 4, 6, 14, 19, 21 and 28). Relative abundances were attributed to each sequence in a given sample to represent real datasets. In sample 1, the most abundant species represented 50% of the reads and the other species had decreasing read abundances with each having half as many sequences as the previous one. For six samples (2 to 7), species relative abundance were based on real samples from both large and small rivers (Cantera et al., 2019; Milhau et al., 2019; Pont et al., 2018) and marine ecosystems (Polanco Fernández et al., 2020). In samples 9 and 25, all fish species (30 and 18, respectively) had equal abundance. For the other samples, abundances were attributed such that some sequences were very abundant and others rare (see Supporting Information Table S1). We simulated amplicons of the 26 to 60 bp 12S mt rRNA region containing the teleo primer. Species composing each sample were selected randomly from a custom reference database of 2,070 sequences of the fish mitochondrial 12S rRNA gene (downloaded from GenBank, on the 23/01/2019). Each simulated assemblage was replicated 12 times, with the same species and abundance composition, to mimic PCR replicates variability. To obtain a dataset similar to those obtained from high-throughput sequencing, we applied an Illumina error model with 98% substitutions and 2% insertions/deletions. A total of 45,000 reads were simulated in each sample replicate, for a total of 15,660,000 reads in the complete dataset. All the Grinder FASTQ files containing the interleaved amplicon sequences and quality scores for each simulated assemblage were concatenated to obtain output files similar to those obtained after a Miseq sequencing of one library of pooled PCR samples. Grinder also produced a text file describing the abundance of each sequence in the simulated replicates for each sample. These files were used as expected species composition and relative abundances to compute sensitivity, F-measure and RMSE on reads abundance using the outputs of each program involved in the comparative analysis.

The input and output data as well as the code written to construct the simulated dataset are available as a GitHub repository at: https://github.com/lmathon/metabarcoding_data_simulation.

2.2. Selected programs and steps

We selected some of the most cited programs for each step through a literature review with the keywords “bioinformatics”, “metabarcoding” and the name of the analysis step (see Supporting Information Table S2 and Method S1). The most cited programs we considered are listed in Supporting Information Table S3. Here, we use the word “program” to define an independent binary or package dedicated to one of the six steps of eDNA metabarcoding analysis. The word “pipeline” refers to a program or set of programs that proceeds to analyzing all the steps from raw read assembly to species identification.

The characteristics of our simulated dataset (Illumina sequencing, mitochondrial 12S mt rRNA gene region, very short amplicons, fish DNA) excluded many programs from our comparisons since they were not compatible with such data. For example, Mothur is specialized in analyzing 16S rRNA gene sequences, TagCleaner and DeML do not support paired-end reads, and Kaiju is specialized in protein-level assignment (see Supporting Information Table S3). QIIME is no longer maintained and some studies have shown that it requires a long execution time (Bonder et al., 2012) and the plugins used give a F-measure worse than that of QIIME2 (Gardner et al., 2019). As a result, QIIME2 was tested instead of QIIME.

USEARCH (R. C. Edgar, 2010) is a widely used program but is only available as open-source in its memory-limited 32-bit version which does not meet our open-source requirement. Instead, we used VSEARCH, its open-source equivalent (Rognes et al., 2016).

To compare and identify the best programs, each bioinformatic step was tested successively by changing the program that performs this part of the analysis and by maintaining all others fixed. We decided to use the OBITools pipeline (Boyer et al., 2016) as a backbone for the performance tests (Fig. 1), because this pipeline generates reliable results for fish eDNA metabarcoding data (Bylemans et al., 2018; Pont et al., 2018; Sales et al., 2019). All OBITools programs composing the fixed pipeline were evaluated in parallel with the other programs tested for each step. The steps were defined as follows: 1) Merging, where forward and reverse reads were aligned to create a single consensus sequence, 2) Demultiplexing, which assigned each sequence to its sample and removed the primers, 3) Dereplication, or keeping only one representing sequence and count for strictly identical sequences, 4) Quality filtering, which removed sequences that were too short or contained ambiguous bases, 5) Removing variants/PCR errors, so taking into account that real haplotypes and variants due to intractable sequencing/PCR errors should be grouped to avoid overestimating species richness and 6) Identifying taxa, where a taxon was assigned to each sequence. For this last step, we used the same reference database as the one used for the Grinder simulations. Only sequences assigned to the species level with more than 98% similarity were considered for species identification. The 10 programs compared in this study are listed in Table 1. Parameters chosen at each step for each program being compared can be found in Table 2. Since Grinder cannot simulate chimeras, the chimera removal step was not tested in this study. Each program was run on a cluster using Ubuntu 18.04.3 LTS with 128GB RAM and 1 CPU to obtain comparable execution times. Data and software commands necessary to reproduce this study are available at: https://github.com/lmathon/eDNA--benchmark_pipelines.

2.3. Performance evaluation

Each program was evaluated by calculating indices that quantify its ability to produce accurate species lists (sensitivity and F-measure) and relative read abundances expected from the known or ground-truth simulated communities (RMSE). The execution time of each program was also recorded. Details on the computation of execution times can be found in Supporting Information Method S2.

After taxonomic assignment, sequence counts were aggregated by species and by replicate. For each tested program, the number of false positives (FP, species present in the output of the program but not in the initial community), true positives (TP, species present in both the output of the program and the initial community) and false negatives (FN, species present in the initial community but not in the output of the program) were calculated. We then computed the sensitivity (equation 1) and F-measure (equation 2) indices for each replicate of each sample from FP, TP and FN, as suggested in Gardner et al. (2019):

$$sensitivity = \frac{TP}{TP + FN} \quad (1)$$

$$F - measure = \frac{2TP}{2TP + FP + FN} \quad (2)$$

These indices present complementary advantages. Sensitivity is relevant to identify programs missing rare taxa while the F-measure highlights programs detecting false positives. For each sample, we derived the mean and the standard error of these indices for each replicate. The standard error among replicates represents intra-sample variability for a given program. The mean and standard error among the 29 samples represent the inter-sample variability for the results of each program. We compared the indices averaged across all samples between programs to determine if there was a significant difference of performance between programs. Since the data were not normally distributed, the mean comparison was carried out with a non-parametric Kruskal-Wallis test followed by a Dunn post-hoc test. Sensitivity and F-measure were also calculated after removing singletons from the dataset (sequences with only 1 read in the pipeline outputs).

Here, we are referring to species abundance as the number of reads assigned to a given species in each sample replicate. For each replicate of each sample, the relative abundance of each species was calculated from the total number of reads and compared to the expected relative abundances for each species (simulated by the Grinder program). The root mean square error (RMSE) was then calculated for each abundance comparison. This index quantifies the level of dissimilarity between two lists of abundances: the lower the RMSE is, the more similar are the observed and expected relative abundances. The mean RMSE per sample was calculated as well as the associated standard error. Sensitivity, F-measure and RMSE were also calculated per sample, after summing the species counts in the twelve replicates. Statistical analyses were carried out with R v3.5.3.

2.4. Comparing assembly of best programs to full pipelines

A complete pipeline was built by assembling the most performant programs for each step based on the performance indices to detect species occurrence, to retrieve the relative read abundances and the execution time. Formatting scripts were written when necessary to facilitate the transition between programs. This pipeline was compared to other pipelines, namely Barque v1.6.2, OBITools v1.2.13 and QIIME2 (Bolyen et al., 2019). Since QIIME2 is a toolbox, the results will be dependent on the plugins used. Here we used demux for demultiplexing, cutadapt for primer removal, DADA2 (Callahan et al., 2016) for error removal, dbotu-q2 for ASV clustering and sklearn-classifier for taxonomic assignment. Barque uses trimmomatic for filtering, Flash for merging, its own python script to split amplicons, and VSEARCH for taxonomic assignment. The steps, programs, and parameters used by Barque and QIIME2-based pipelines can be found in Table 3. Because Barque takes demultiplexed reads as inputs, the demultiplexing was performed upstream with Cutadapt using the same parameters as for our assembled pipeline. Each pipeline was run using 16 CPUs. The same performance indices (sensitivity, F-measure and RMSE on reads abundance) were calculated for the outputs of each pipeline and compared.

2.5. Illustrating the benchmark on real data

The same comparison process was run on an empirical dataset obtained from the Mediterranean Sea. The eDNA samples were collected on the 5th of June 2018 in four replicates within the no-take reserve of Carry-le-Rouet, at 5 km, and at 10 km outside the reserve, for a total of 12 samples (Boulanger et al., 2021). For each sample, 30 L of seawater were continuously collected along a 2km transect from approximately 1m below the surface. Transects were conducted close to the coastline and the substrate to ensure the sampling of coastal organisms. Seawater samples were filtered on site using a VigiDNA® 0.2 µM cross flow filtration capsule (SPYGEN, le Bourget du Lac, France). Immediately after filtration, the capsule was drained by filtering air, filled with 80 ml of CL1 buffer (SPYGEN) and stored at room temperature until the extraction. DNA extraction, amplification (12 replicates per sample) and sequencing followed the protocol described in Polanco Fernández et al., (2020). The different programs and pipelines were run on the raw sequences obtained after sequencing. The reference database used for the taxonomic assignment was built by performing in-silico PCR with teleo primers using ecoPCR (Boyer et al., 2016) on the entire public database ENA (Leinonen et al., 2011; release 141) and by adding sequences from Mediterranean species sequenced by our group (Boulanger et al., 2021). Since the information about actual read abundances in the environment is unknown, it was not possible to measure the RMSE index. Hence, only the sensitivity and F-measure indices were measured before and after removing singletons in samples. To do so, fish species lists obtained by each program or pipeline were compared to lists of fish species identified by underwater visual census in Carry-le-Rouet reserve and outside, during several campaigns in 2018 (Charbonnel, Monin, & Bachet, 2020). Those lists obtained by independent sampling methods were considered as the expected species occurrences. To measure comparable execution time, each individual program was run using 1 CPU, and each pipeline was run using 16 CPUs.

3. Results

3.1. Sensitivity, F-measure and RMSE on abundances

For each program tested, a mean index was estimated by averaging raw values of indices across replicates and samples.

For the merging, demultiplexing, dereplication, read filtering and error removal steps, the sensitivity, F-measure and RMSE were not significantly different between the programs (Supporting Information Fig S1-3). The mean sensitivity obtained with the full OBITools pipeline was 0.94 and ranged from 0.78 to 1 with a mean standard error per sample of 0.004. The mean F-measure obtained with OBITools was 0.97 (ranged between 0.88 and 1). The mean RMSE between the relative abundances obtained for each replicate and the expected relative abundances was 1.1 with OBITools (ranged between 0.09 and 4.6).

We found significant differences between programs only for the taxonomic assignment step (Fig. 2). Taxonomic assignment with Sintax produced significantly lower sensitivity (0.57, Fig. 2a, p=4.8e-13) and F-measure (0.71, Fig. 2b, p=7.1e-13) and higher RMSE (3.2, Fig. 2c, p=2e-08) than VSEARCH –*usearch_global* and *ecotag*. VSEARCH –*usearch_global* provided a significantly higher mean sensitivity than *ecotag* (0.97), and significantly lower mean RMSE (0.4). The assignment program VSEARCH was therefore more accurate when evaluating community composition and read abundances than *ecotag*.

Sensitivity and F-measures, after removing singletons, showed the same pattern but were lower due to less TP and more FN (Supporting Information Fig. S4-5). Sensitivity and F-measure per sample were higher and RMSE lower due to the increased detection of rare species when pooling the 12 replicates (Supporting Information Fig. S6-8), but the difference between programs showed the same patterns.

3.2. Execution time

Execution time varied importantly between programs (Fig. 3). For all but one step, OBITools programs were the slowest, sometimes by a factor of more than 200. The fastest program for merging was VSEARCH (3.1 min, Fig. 3). Demultiplexing with Cutadapt was faster than with *ngsfilter* (30 min and 488 min respectively, Fig. 3). Execution time of the sequence dereplication step was 198 min with *obiuniq*, and 0.8 min only with VSEARCH (Fig. 3). VSEARCH and Flexbar were faster than other programs to filter reads (0.2 min). Execution time of the PCR and sequencing error removal step lasted 17.4 min with *obiclean*, while Swarm and VSEARCH –*cluster_unoise* ran in 0.4 min (Fig. 3). Sintax and VSEARCH –*usearch_global* executed the assignment in 1.8 and 0.14 min respectively while *ecotag* (OBITools) ran in 58 min on our simulated dataset (Fig. 3).

3.3. Comparison between pipelines

From the step comparison results between programs, we selected the best ones in terms of sensitivity, F-measure, RMSE on the abundance, and execution time. Since indices varied in the same direction, the selection was straightforward. These selected programs were integrated in a pipeline following the order of the steps shown in Figure 1. This custom pipeline was composed of VSEARCH –*fastq_mergepairs* for assembling the reads, Cutadapt for demultiplexing, VSEARCH –*derep_fulllength* for the dereplication, VSEARCH –*fastx_filter* for the quality filtering, Swarm for the suppression of PCR and sequencing errors and VSEARCH –*usearch_global* for the taxonomic assignment.

Our assembled pipeline obtained a mean sensitivity of 0.97, the same as Barque (0.97), higher than OBITools (0.94) and significantly higher than QIIME2-based (0.9) (Fig. 4a, p=6.4e-03). The mean F-measure was the same for the assembled pipeline and Barque (0.98) and significantly higher than OBITools (0.97, Fig. 4b, p=0.05) and QIIME2-based (0.94). Barque and our assembled pipeline were also the best pipelines to recover relative abundances, and mean RMSE were not significantly different, with 0.31 for Barque and 0.44 for our pipeline, while mean RMSE were significantly higher for OBITools (1.1) and QIIME2-based (1.24) (Fig. 4c, p=2.5e-07). Sensitivity and F-measures after removing singletons showed the same pattern but were lower due to less TP and more FN (Supporting Information Fig. S9).

The execution times of the four pipelines were very different. Barque alone ran in 2min25sec and in 30 min when the demultiplexing with Cutadapt was added. With 16 CPUs used where possible, our assembled pipeline ran in 46 min and QIIME2 in 95 min. OBITools was the longest and ran in 1010 min so 40 times slower than Barque (Fig. 4d).

The percentage of reads assigned to the species level with 98% similarity also differed between pipelines. Barque was able to assign a species name to 98.7% of the raw demultiplexed reads (15,458,570) whereas our pipeline assigned 95.6% of the reads (14,970,256 reads), OBITools 94.4% (14,783,635), and QIIME2 91.5% (14,316,059).

3.4. Illustration from real data

The comparison of the program performances on empirical data provided results identical to those obtained with the simulated dataset. The only significant difference was found for the assignment step where Sintax obtained a significantly lower F-measure. The sensitivity and F-measure showed slight variations between programs, but these were not significant (Supporting Information Fig. S10-11). After removing singletons, sensitivity and F-measures showed the same variation which were lower due to less TP and more FN (Supporting Information Fig. S12-13). Execution time on real data confirmed that VSEARCH was the fastest program for merging, dereplicating, filtering and assigning (along with Sintax), Cutadapt was fastest for demultiplexing, and Swarm was fastest for cleaning errors (Supporting Information Fig. S14). The performance comparison between our assembled pipeline and the other pipelines provided results concordant with the analyses on simulated data. QIIME2-based pipeline was significantly less performant than Barque, OBITools and our assembled pipeline for sensitivity (Fig. 5a, p=5.7e-06) and F-measure (Fig. 5b, p=2.2e-06), also when removing singletons (Fig. 5c-d). Barque was only significantly less performant than OBITools and the assembled pipeline for F-measure, due to a slightly higher number of FP. Execution times were much shorter for Barque and the assembled pipeline (53 and 155 minutes respectively, Fig. 5e).

The lower sensitivity and F-measure values obtained with all programs tested on the empirical dataset were due to a high number of FN (species seen by divers and not found with eDNA). However, many of these species were identified with eDNA with a similarity to the reference sequence that was lower than our threshold of 98% and were thus discarded from further analyses.

4. Discussion

4.1. A step-by-step comparison between programs

The results of our program comparison allowed us to select the best programs for retrieving the initial community composition and abundance structure of both simulated and real fish communities. For five out of six steps, execution time was the most discriminant factor. OBITools programs obtain high sensitivities and F-measures but require much longer execution times than the other programs. Results obtained with the real dataset are similar to those obtained with the simulated data. The assignment step was the only one showing significant differences between programs indices, and time was the deciding factor for all other steps.

We thus provide recommendations for programs to use at each step. For merging reads, we recommend to use VSEARCH *–fastq_mergepairs*, which is the fastest program. For demultiplexing we suggest Cutadapt, which has similar performance as OBITools' *ngsfilter* but is much faster. VSEARCH *–derep_fulllength* and --*fastx_filter* are retained for dereplication and read filtering respectively, because they obtain similar performances as other programs tested, but are faster. For error removal, we recommend using Swarm, which produces results as good as obiclean and VSEARCH --*cluster_unoise* but is faster. VSEARCH –*usearch_global* provides significantly better results than Sintax and *ecotag* for taxonomic assignment with a complete reference database, both in terms of sensitivity, F-measure and RMSE on relative abundances.

4.2. Comparison of complete pipelines

The comparison of complete pipelines shows that Barque obtains sensitivity and F-measures as high as those of the assembled pipeline made of the best individual programs. These two pipelines also report the most accurate estimates of relative abundances. Barque is the fastest pipeline while our pipeline takes 1.5 times longer to analyse the same simulated dataset. QIIME2-based pipeline is slightly slower than our pipeline while taxonomic and abundance results are significantly worse than with the other pipelines. OBITools requires more than 30 times the running time of Barque, the fastest pipeline. It also returns significantly worse results for abundances RMSE but provides good sensitivity and F-measures. It is noteworthy to consider that the intentions behind the design of these pipelines differ. For example, Barque aims to be exhaustive in the detection of species while minimizing the risk of not detecting a rare species of potential interest, such as an invasive species, so omits a denoising step. As a result, Barque annotates a higher proportion of the raw reads, and also produces slightly more false positives than our pipeline. In contrast, while the goal of our pipeline is also to provide species abundances that are as close to reality as possible, it controls more stringently for false positives by using a denoising step, which could lead to the removal of some very rare species and to less annotated reads. Despite these different designs, Barque and our pipeline give almost identical results on all three indices. QIIME2 is a toolbox with many different steps where several plugins are available. The plugins chosen in the QIIME2-based pipeline were the most suitable for our data given the goal of our study, and the most comparable to the tools comprised in the other pipelines. However, many other possibilities exist and choosing other plugins and other treatment steps would result in as many different pipelines, each with a different outcome. QIIME2 offers numerous possibilities, and choosing the most appropriate tools for the purpose of a given study is important as this will influence the results and interpretation.

Analyses on the empirical dataset also revealed that the new pipeline is the most performant. Barque appears slightly less performant on the real data due to an important number of what was classified as false positives in the dataset. However, many of these species were observed by divers in different years. These species were thus probably present in the area during eDNA sampling even though they were not spotted by divers at the time of their campaign. As a result, it is important to be critical towards the species identified by eDNA and keep in mind that they could be real occurrences even if they were not reported using conventional observation methods. Therefore, in the context of Teleostean metabarcoding based on primer teleo, we recommend using Barque or the assembled pipeline. However, it is important to keep in mind that each pipeline considered here is a bespoke solution to the questions we aimed to address. Moreover, the differences observed in the performance of each pipeline depends on the choice of tools composing each pipeline.

4.3. Taxonomic assignment

The three taxonomic assignment algorithms we tested differ in many aspects. OBITools's *ecotag* searches the reference database to find the reference sequence with the highest similarity to the query sequence. It then searches for all other potential reference sequences with a similarity to the first reference sequence equal or higher than the similarity to the query sequence. *Ecotag* then assigns the query sequence to the lowest common ancestor (LCA) of reference sequences. *Ecotag* provides a taxon name at the family, genus or species level as well as information about all matching reference sequences (Boyer et al., 2016).

VSEARCH *–usearch_global* filters reference sequences that share the highest number of k-mers with the query sequence and then computes the optimal global alignment between the query sequence and these reference sequences (Rognes et al., 2016). The taxonomic assignment contains the list of species matching to the query sequence with equal similarity. *Sintax* also proceeds with a k-mers search, among 100 iterations (Edgar, 2016). For each iteration, a sub-sample of k-mers contained in the query sequence is extracted. The reference sequence that has the maximum k-mers in common is retained and the taxonomy is taken from this sequence. After the 100 iterations, the species name that occurs most often is identified and its frequency is reported, along with the frequency of the genus and family identifications. If these frequencies are lower than the chosen identity threshold, then the assignment is not retained. QIIME2-based pipeline uses the plugin classify-sklearn that apply a machine learning classifier from the SciKit-learn algorithm (Pedregosa et al., 2011). The method is based on k-mer counts extraction from the reference sequences (up to 32-mers) and training of the scikit-learn multinomial naive Bayes classifier (Bokulich et al., 2018). Barque also uses VSEARCH *--usearch_global* and provides assignments to species, genus, and group level (which can be anything above the genus, for example the family) and uses different similarity thresholds for assignment at each of these three levels. However, assignments to taxonomic levels above the species level were not analyzed here, as we focused on species name assignments.

4.4. Sources of variation in species detection

Some samples from the simulated dataset obtained much poorer sensitivity and F-measure values regardless of the program used. This is due to the presence of false negatives of several origins. First, some species of the custom reference database used for simulation and assignment have 100% identical sequences for this portion of the 12S rRNA gene and are therefore not distinguishable at the species level (e.g. *Neosalanx taihuensis* and *N. tangkahkeii*). Secondly, some species have a very low abundance and are therefore not found in each of the 12 replicates of the samples in the pipeline outputs. The presence of these false negatives also influences the RMSE between the expected and obtained abundances. For each of these false negatives, the observed relative abundance is 0 and the RMSE is thus higher for these replicates. The computation of the performance indices per sample, after pooling the observations in the 12 replicates, shows better results, as the full consideration of rare species decreases the number of false negatives. Only a few false positives are observed in the outputs of the different programs and pipelines; they are due to wrong taxonomic assignments either caused by sequencing errors introduced during the amplicon simulations or represent residual errors in the real dataset.

Performance indices for the empirical dataset are lower than the ones on the simulated data due to a high number of false negatives (some due to the too stringent similarity threshold) and some false positives. It is likely that with an eDNA sampling limited to one day, we did not detect all of the species present on site and that divers did not detect elusive or hidden species (Aglieri et al., 2020; Polanco F. et al., 2020). In order to reduce the number of false negatives and false positives, it would be necessary to extend the eDNA sampling of each site to several seasons and to consider similarity thresholds adapted for the taxonomic assignment.

For all programs tested in this study, as well as for both datasets, we looked at the impact of the removal of singletons. This results in a slight decrease in sensitivity and F-measure (see Supporting Information Fig. S4-6 and S9-10). After removing the sequences with a very weak representation, some of the false positives are removed. However, removing the singletons also removes species with real but very low abundances, thus increasing the number of false negatives, which can also lead to bad interpretations of species presence/absence. In real eDNA samples, singletons represent rare taxa of high interest, like invasive or threatened species, but also contamination and PCR or sequencing errors, such as tag or index jumps (Kwok & Higuchi, 1989; Schnell et al., 2015; Taberlet et al., 2018). In the real data from Carry-le-Rouet, the removal of singletons led to the loss of true positives, indicating that eDNA can detect rare and low-abundance species. The decision to remove singletons from a dataset should then depend on the objectives and preferences of the study, or aimed at finding a balance between removing all contaminations and errors and retaining higher chances to detect rare species.

4.5. Perspectives

In this study we focused on a specific fish 12S mitochondrial gene region but this benchmark process could be extended to other taxa and barcodes with only slight modifications. When using different markers, depending for example on the size of the barcode and the completeness of reference databases, some parameters will have to be updated, but the general bioinformatic treatment will be similar, and the same programs can be used.

The same comparison could be extended to recent bioinformatic programs, or programs not considered in this study, such as MeFit for merging and filtering (Parikh et al., 2016), or DUDE-seq for denoising and correction of sequencing errors (Lee et al., 2017). We could also apply our comparison approach on other pipelines, such as eDNAFlow which produces ZOTUs and uses a LCA assignment method (Mousavi-Derazmahalleh et al., 2021) or CoMa (Hupfauf et al., 2020).

The similarity threshold set at 98% for assigning a sequence to a species is equivalent, on our short amplicons, to a maximum of either zero or one mismatch between the query and the reference sequences, depending on the length of the amplicon, which varies from one species to another. This can result, as in the case of the real data analyzed here, in the removal of a number of sequences that would have been correctly assigned with a lower confidence and thus lead to some false negatives. Therefore, it would be relevant to consider adaptive thresholds.

In this study, we focused on taxonomic assignment at the species level. As a result, we did not explore the ability of the algorithm to provide taxonomic assignment above the species level. Nevertheless, it would be worthwhile for ecological applications to consider higher taxonomic assignment using an algorithm with such abilities, especially in the case of incomplete reference databases. PROTAX, for example, is a probabilistic method for taxonomic assignment that uses outputs of other classifiers (BLAST, RDP classifier, Wang et al., 2007) as predictors (Somervuo et al., 2016). The Anacapa Toolkit (Curd et al., 2019) includes the Anacapa Classifier module that aligns ASVs to a reference database using Bowtie2 and assigns taxonomy with a Bayesian Lowest Common Ancestor (BLCA) method (Gao et al., 2017). These two approaches might provide relevant results for taxonomic assignments at higher levels, with probability and confidence scores.

Conclusion

The main finding of this study is that the choice of a given program for eDNA metabarcoding analysis depends mostly on the taxonomic assignment step and the resulting diversity estimates. For all other steps, the only difference between programs standardized with the same parameters is in the execution time. This study provides some guidance for the choice of the best bioinformatics tools or the best pipeline to use for analysis of eDNA metabarcoding data. Most importantly, this study highlights the need for more efficient and accurate tools for eDNA metabarcoding taxonomic assignments, especially when only incomplete reference databases are available.

Acknowledgements

This study was part of the ALIVe project (funding PIA / ADEME / SPYGEN). The simulation and program comparison algorithms benefited from the Montpellier Bioinformatics Biodiversity platform supported by the LabEx CeMEB, an ANR "Investissements d'avenir" program (ANR-10-LABX-04-01). WT and CL received funding from the French Agence Nationale de la Recherche (ANR) through the GlobNets (ANR-16-CE02-0009) project and from 'Investissement d'Avenir' grants managed by the ANR (Trajectories: ANR-15-IDEX-02; Montane: OSUG@2020: ANR-10-LAB-56). We are grateful for the discussions and advice given by Lucie Zinger and Frédéric Boyer. We also thank Lucie Zinger for the comments and suggestions on the manuscript. Finally we are grateful to Benjamin Sibbet and Carla Lopes and anonymous referees for their constructive comments on a previous version of this manuscript.

Data accessibility

The input data used for the simulated and real analysis are available on Dryad at https://doi.org/10.5061/dryad.15dv41nx6. The code for the simulation protocol, inputs and outputs are accessible at https://github.com/lmathon/metabarcoding_data_simulation. The codes for the benchmark study can be found at: github.com/lmathon/eDNA--benchmark_pipelines. The version of BARQUE used in this paper is available at: https://github.com/enormandeau/barque/releases/tag/v1.6.2.

Author contributions

LM, AV, CL, SM and TD designed the experiment. LM and EN simulated the data. LM, PEG, EN, CN and CL ran the programs. CL, PEG and LM wrote the formatting scripts. LM analyzed the data and wrote the manuscript. All authors contributed critically to the drafts and gave final approval for publication.

References

Aglieri, G., Baillie, C., Mariani, S., Cattano, C., Calò, A., Turco, G., … Milazzo, M. (2020). Environmental DNA effectively captures functional diversity of coastal fish communities. *Molecular Ecology*, (August), 1–13. https://doi.org/10.1111/mec.15661

Angly, F. E., Willner, D., Rohwer, F., Hugenholtz, P., & Tyson, G. W. (2012). Grinder: A versatile amplicon and shotgun sequence simulator. *Nucleic Acids Research*, *40*(12). https://doi.org/10.1093/nar/gks251

Bazinet, A. L., & Cummings, M. P. (2012). A comparative evaluation of sequence classification programs. *BMC Bioinformatics*, *13*(1). https://doi.org/10.1186/1471-2105-13-92

Berger, C. S., Hernandez, C., Laporte, M., Côté, G., Paradis, Y., Kameni T., D. W., … Bernatchez, L. (2020). Fine-scale environmental heterogeneity shapes fluvial fish communities as revealed by eDNA metabarcoding. *Environmental DNA*, *2*(4), 647–666. https://doi.org/10.1002/edn3.129

Bokulich, N. A., Kaehler, B. D., Rideout, J. R., Dillon, M., Bolyen, E., Knight, R., … Gregory Caporaso, J. (2018). Optimizing taxonomic classification of marker-gene amplicon sequences with QIIME 2's q2-feature-classifier plugin. *Microbiome*, *6*(1), 1–17. https://doi.org/10.1186/s40168-018-0470-z

Bolyen, E., Rideout, J. R., Dillon, M. R., Bokulich, N. A., Abnet, C., Ghalith, G. A. Al, … Naimey, A. T. (2019). QIIME 2 : Reproducible , interactive , scalable , and extensible microbiome data science. *Nature Biotechnology*, *32*, 852–857. https://doi.org/10.7287/peerj.preprints.27295

Bonder, M. J., Abeln, S., Zaura, E., & Brandt, B. W. (2012). Comparing clustering and pre-processing in taxonomy analysis. *Bioinformatics*, *28*(22), 2891–2897.

Boulanger, E., Loiseau, N., Valentini, A., Arnal, V., Boissery, P., Dejean, T., … Mouillot, D. (2021). Environmental DNA metabarcoding reveals and unpacks a biodiversity conservation paradox in Mediterranean marine reserves, Dryad, Dataset. *Proceedings of the Royal Society B*, *288*(20210112).

Boyer, F., Mercier, C., Bonin, A., Le Bras, Y., Taberlet, P., & Coissac, E. (2016). obitools: A unix-inspired software package for DNA metabarcoding. *Molecular Ecology Resources*, *16*(1), 176–182. https://doi.org/10.1111/1755-0998.12428

Bylemans, J., Furlan, E. M., Gleeson, D. M., Hardy, C. M., & Duncan, R. P. (2018). Does Size Matter? An Experimental Evaluation of the Relative Abundance and Decay Rates of Aquatic Environmental DNA. *Environmental Science and Technology*, *52*(11), 6408–6416. https://doi.org/10.1021/acs.est.8b01071

Calderón-Sanou, I., Münkemüller, T., Boyer, F., Zinger, L., & Thuiller, W. (2020). From environmental DNA sequences to ecological conclusions: How strong is the influence of methodological choices? *Journal of Biogeography*, (July), 1–14. https://doi.org/10.1111/jbi.13681

Callahan, B. J., McMurdie, P. J., Rosen, M. J., Han, A. W., Johnson, A. J. A., & Holmes, S. P. (2016). DADA2: High-resolution sample inference from Illumina amplicon data. *Nature Methods*, *13*(7), 581–583. https://doi.org/10.1038/nmeth.3869

Cantera, I., Cilleros, K., Valentini, A., Cerdan, A., Dejean, T., Iribar, A., … Brosse, S. (2019). Optimizing environmental DNA sampling effort for fish inventories in tropical streams and rivers. *Scientific Reports*, *9*(1), 1–11. https://doi.org/10.1038/s41598-019-39399-5

Charbonnel, E., Monin, M., & Bachet, F. (2020). *Suivi des peuplements de poissons de la réserve marine de Carry-le-Rouet (Parc Marin de la Côte Bleue) – Bilan années 2011-2020.*

Civade, R., Dejean, T., Valentini, A., Roset, N., Raymond, J. C., Bonin, A., … Pont, D. (2016). Spatial Representativeness of Environmental DNA Metabarcoding Signal for Fish Biodiversity Assessment in a Natural Freshwater System. *PLoS ONE*, *11*(6), 1–19. https://doi.org/10.1371/journal.pone.0157366

Collins, R. A., Bakker, J., Wangensteen, O. S., Soto, A. Z., Corrigan, L., Sims, D. W., … Mariani, S. (2019). Non-specific amplification compromises environmental DNA metabarcoding with COI. *Methods in Ecology and Evolution*, *10*(11), 1985–2001. https://doi.org/10.1111/2041-210X.13276

Curd, E. E., Gold, Z., Kandlikar, G. S., Gomer, J., Ogden, M., O'Connell, T., … Meyer, R. S. (2019). Anacapa Toolkit: An environmental DNA toolkit for processing multilocus metabarcode datasets. *Methods in Ecology and Evolution*, *10*(9), 1469–1475. https://doi.org/10.1111/2041-210X.13214

Deiner, K., Bik, H. M., Mächler, E., Seymour, M., Lacoursière-Roussel, A., Altermatt, F., … Bernatchez, L. (2017). Environmental DNA metabarcoding: Transforming how we survey animal and plant communities. *Molecular Ecology*, *26*(21), 5872–5895. https://doi.org/10.1111/mec.14350

Deiner, K., Yamanaka, H., & Bernatchez, L. (2020). The future of biodiversity monitoring and conservation utilizing environmental DNA. *Environmental DNA*, (December 2020), 1–5. https://doi.org/10.1002/edn3.178

Djurhuus, A., Closek, C. J., Kelly, R. P., Pitz, K. J., Michisaki, R. P., Starks, H. A., … Breitbart, M. (2020). Environmental DNA reveals seasonal shifts and potential interactions in a marine community. *Nature Communications*, *11*(254), 1–9. https://doi.org/10.1038/s41467-019-14105-1

Edgar, R. (2016). SINTAX: a simple non-Bayesian taxonomy classifier for 16S and ITS sequences.

BioRxiv, 1–20. https://doi.org/10.1101/074161

Edgar, R. C. (2010). Search and clustering orders of magnitude faster than BLAST. *Bioinformatics*, *26*(19), 2460–2461. https://doi.org/10.1093/bioinformatics/btq461

Gao, X., Lin, H., Revanna, K., & Dong, Q. (2017). A Bayesian taxonomic classification method for 16S rRNA gene sequences with improved species-level accuracy. *BMC Bioinformatics*, *18*(1), 1–10. https://doi.org/10.1186/s12859-017-1670-4

Gardner, P. P., Watson, R. J., Stott, M. B., Morales, S. E., Morgan, X. C., Finn, R. D., & Draper, J. L. (2019). Identifying accurate metagenome and amplicon software via a meta-analysis of sequence to taxonomy benchmarking studies. *PeerJ*, *7*, 1–19. https://doi.org/10.7717/peerj.6160

Hupfauf, S., Etemadi, M., Juárez, M. F. D., Gómez-Brandón, M., Insam, H., & Podmirseg, S. M. (2020). CoMA – an intuitive and user-friendly pipeline for amplicon-sequencing data analysis. *PLoS ONE*, *15*(12 December), 1–28. https://doi.org/10.1371/journal.pone.0243241

Jerde, C. L., Wilson, E. A., & Dressler, T. L. (2019). Measuring global fish species richness with eDNA metabarcoding. *Molecular Ecology Resources*, *19*(1), 19–22. https://doi.org/10.1111/1755-0998.12929

Juhel, J. B., Utama, R. S., Marques, V., Vimono, I. B., Sugeha, H. Y., Kadarusman, … Hocdé, R. (2020). Accumulation curves of environmental DNA sequences predict coastal fish diversity in the coral triangle. *Proceedings. Biological Sciences*, *287*(1930), 1–10. https://doi.org/10.1098/rspb.2020.0248

Kwok, S., & Higuchi, R. (1989). Avoiding false positives with PCR. *Nature*, *339*(6221), 237–238. https://doi.org/10.1038/339237a0

Lee, B., Moon, T., Yoon, S., & Weissman, T. (2017). DudE-Seq: Fast, flexible, and robust denoising for targeted amplicon sequencing. *PLoS ONE*, *12*(7), 1–25. https://doi.org/10.1371/journal.pone.0181463

Leinonen, R., Akhtar, R., Birney, E., Bower, L., Cerdeno-Tárraga, A., Cheng, Y., … Cochrane, G. (2011). The European nucleotide archive. *Nucleic Acids Research*, *39*(SUPPL. 1), 44–47. https://doi.org/10.1093/nar/gkq967

Lindgreen, S., Adair, K. L., & Gardner, P. P. (2016). An evaluation of the accuracy and speed of metagenome analysis tools. *Scientific Reports*, *6*, 1–14. https://doi.org/10.1038/srep19233

Marques, V., Guérin, P. É., Rocle, M., Valentini, A., Manel, S., Mouillot, D., & Dejean, T. (2020). Blind assessment of vertebrate taxonomic diversity across spatial scales by clustering environmental DNA metabarcoding sequences. *Ecography*, *43*, 1–12.

McElroy, M. E., Dressler, T. L., Titcomb, G. C., Wilson, E. A., Deiner, K., Dudley, T. L., … Jerde, C. L. (2020). Calibrating Environmental DNA Metabarcoding to Conventional Surveys for Measuring Fish Species Richness. *Frontiers in Ecology and Evolution*, *8*, 0–12. https://doi.org/10.3389/fevo.2020.00276

Milhau, T., Valentini, A., Poulet, N., Roset, N., Jean, P., Gaboriaud, C., & Dejean, T. (2019). Seasonal dynamics of riverine fish communities using eDNA. *J Fish Biol. Accepted Author Manuscript.*, 1–36. https://doi.org/10.1111/1744-1633.12020

Minamoto, T., Yamanaka, H., Takahara, T., Honjo, M. N., & Kawabata, Z. (2012). Surveillance of fish species composition using environmental DNA. *Limnology*, *13*(2), 193–197. https://doi.org/10.1007/s10201-011-0362-4

Mousavi-Derazmahalleh, M., Stott, A., Lines, R., Peverley, G., Nester, G., Simpson, T., … Christophersen, C. T. (2021). eDNAFlow, an automated, reproducible and scalable workflow for analysis of environmental DNA (eDNA) sequences exploiting Nextflow and Singularity. *Molecular Ecology Resources*, (August 2020), 1–8. https://doi.org/10.1111/1755-0998.13356

Parikh, H. I., Koparde, V. N., Bradley, S. P., Buck, G. A., & Sheth, N. U. (2016). MeFiT: Merging and filtering tool for illumina paired-end reads for 16S rRNA amplicon sequencing. *BMC Bioinformatics*, *17*(1), 1–6. https://doi.org/10.1186/s12859-016-1358-1

Pauvert, C., Buée, M., Laval, V., Edel-Hermann, V., Fauchery, L., Gautier, A., … Vacher, C. (2019). Bioinformatics matters: The accuracy of plant and soil fungal community data is highly dependent on the metabarcoding pipeline. *Fungal Ecology*, *41*, 23–33. https://doi.org/10.1016/j.funeco.2019.03.005

Pawlowski, J., Kelly-Quinn, M., Altermatt, F., Apothéloz-Perret-Gentil, L., Beja, P., Boggero, A., … Kahlert, M. (2018). The future of biotic indices in the ecogenomic era: Integrating (e)DNA metabarcoding in biological assessment of aquatic ecosystems. *Science of the Total Environment*, *637–638*, 1295–1310. https://doi.org/10.1016/j.scitotenv.2018.05.002

Peabody, M. A., Van Rossum, T., Lo, R., & Brinkman, F. S. L. (2015). Evaluation of shotgun metagenomics sequence classification methods using in silico and in vitro simulated communities. *BMC Bioinformatics*, *16*(1). https://doi.org/10.1186/s12859-015-0788-5

Pedregosa, F., Varoquaux, G., Thirion, B., Gramfort, A., Michel, V., Grisel, O., … Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Polanco F., A., Fopp, F., Albouy, C., Brun, P., Boschman, L., & Pellissier, L. (2020). Marine fish diversity in Tropical America associated with both past and present environmental conditions. *Journal of*

Biogeography, (August), 2597–2610. https://doi.org/10.1111/jbi.13985

Polanco Fernández, A., Marques, V., Fopp, F., Juhel, J., Borrero-Pérez, G. H., Cheutin, M., … Pellissier, L. (2020). Comparing environmental DNA metabarcoding and underwater visual census to monitor tropical reef fishes. *Environmental DNA*, 1–15. https://doi.org/10.1002/edn3.140

Pont, D., Rocle, M., Valentini, A., Civade, R., Jean, P., Maire, A., … Dejean, T. (2018). Environmental DNA reveals quantitative patterns of fish biodiversity in large rivers despite its downstream transportation. *Scientific Reports*, *8*(1), 1–13. https://doi.org/10.1038/s41598-018-28424-8

Pont, D., Valentini, A., Rocle, M., Delaigue, O., Jean, P., & Dejean, T. (2019). The future of fish-based ecological assessment of European rivers : from traditional EU Water Framework Directive compliant methods to eDNA metabarcoding-based approaches. *J Fish Biol. Accepted Author Manuscript.*, 1–50. https://doi.org/10.1111/1744-1633.12020

Prodan, A., Tremaroli, V., Brolin, H., Zwinderman, A. H., Nieuwdorp, M., & Levin, E. (2020). Comparing bioinformatic pipelines for microbial 16S rRNA amplicon sequencing. *PLoS ONE*, *15*(1), 1–19. https://doi.org/10.1371/journal.pone.0227434

Robson, H. L. A., Noble, T. H., Saunders, R. J., Robson, S. K. A., Burrows, D. W., & Jerry, D. R. (2016). Fine-tuning for the tropics: application of eDNA technology for invasive fish detection in tropical freshwater ecosystems. *Molecular Ecology Resources*, *16*(4), 922–932. https://doi.org/10.1111/1755-0998.12505

Rognes, T., Flouri, T., Nichols, B., Quince, C., & Mahé, F. (2016). VSEARCH: a versatile open source tool for metagenomics. *PeerJ*, *4*, 1–22. https://doi.org/10.7717/peerj.2584

Sales, N. G., Wangensteen, O. S., Carvalho, D. C., & Mariani, S. (2019). Influence of preservation methods, sample medium and sampling time on eDNA recovery in a neotropical river. *Environmental DNA*, (April), 119–130. https://doi.org/10.1002/edn3.14

Schnell, I. B., Bohmann, K., & Gilbert, M. T. P. (2015). Tag jumps illuminated - reducing sequence-to-sample misidentifications in metabarcoding studies. *Molecular Ecology Resources*, *15*(6), 1289–1303. https://doi.org/10.1111/1755-0998.12402

Sczyrba, A., Hofmann, P., Belmann, P., Koslicki, D., Janssen, S., Dröge, J., … McHardy, A. C. (2017). Critical Assessment of Metagenome Interpretation - A benchmark of metagenomics software. *Nature Methods*, *14*(11), 1063–1071. https://doi.org/10.1038/nmeth.4458

Siegwald, L., Touzet, H., Lemoine, Y., Hot, D., Audebert, C., & Caboche, S. (2017). Assessment of common and emerging bioinformatics pipelines for targeted metagenomics. *PLoS ONE*, *12*(1), 1–26.

Sigsgaard, E. E., Nielsen, I. B., Carl, H., Krag, M. A., Knudsen, S. W., Xing, Y., … Thomsen, P. F. (2017). Seawater environmental DNA reflects seasonality of a coastal fish community. *Marine Biology*, *164*(6), 1–15. https://doi.org/10.1007/s00227-017-3147-4

Somervuo, P., Koskela, S., Pennanen, J., Henrik Nilsson, R., & Ovaskainen, O. (2016). Unbiased probabilistic taxonomic classification for DNA barcoding. *Bioinformatics*, *32*(19), 2920–2927. https://doi.org/10.1093/bioinformatics/btw346

Taberlet, P., Bonin, A., Coissac, E., & Zinger, L. (2018). *Environmental DNA: For biodiversity research and monitoring*. Retrieved from https://books.google.fr/books?hl=fr&lr=&id=1e9IDwAAQBAJ&oi=fnd&pg=PP1&dq=taberlet+et+al+2018&ots=UX8Vj4tfnO&sig=saiG_Z_TcrrzgtDDsKWizkCwYwc

Thomsen, P. F., Kielgast, J., Iversen, L. L., Møller, P. R., Rasmussen, M., & Willerslev, E. (2012). Detection of a Diverse Marine Fish Fauna Using Environmental DNA from Seawater Samples. *PLoS ONE*, *7*(8), 1–9.

Tsuji, S., Takahara, T., Doi, H., Shibata, N., & Yamanaka, H. (2019). The detection of aquatic macroorganisms using environmental DNA analysis—A review of methods for collection, extraction, and detection. *Environmental DNA*, *1*(2), 99–108. https://doi.org/10.1002/edn3.21

Valentini, A., Taberlet, P., Miaud, C., Civade, R., Herder, J., Thomsen, P. F., … Dejean, T. (2016). Next-generation monitoring of aquatic biodiversity using environmental DNA metabarcoding. *Molecular Ecology*, *25*(4), 929–942. https://doi.org/10.1111/mec.13428

Wang, Q., Garrity, G. M., Tiedje, J. M., & Cole, J. R. (2007). Naïve Bayesian classifier for rapid assignment of rRNA sequences into the new bacterial taxonomy. *Applied and Environmental Microbiology*, *73*(16), 5261–5267. https://doi.org/10.1128/AEM.00062-07

Yan, H. F., Kyne, P. M., Jabado, R. W., Leeney, R. H., Davidson, N. K., Derrick, D. H., … Dulvy, N. K. (2021). Overfishing and habitat loss drives range contraction of iconic marine fishes to near extinction. *Science Advances*, *in press.*, 1–11.

Zinger, L., Bonin, A., Alsos, I. G., Bálint, M., Bik, H., Boyer, F., … Taberlet, P. (2019). DNA metabarcoding—Need for robust experimental designs to draw sound ecological conclusions. *Molecular Ecology*, *28*(8), 1857–1862. https://doi.org/10.1111/mec.15060

Figures and Tables

Table 1. Bioinformatic programs and pipelines used for the comparison

Program	**Step**	**Reference**	**Source code**
OBITools	Pipeline	Boyer et al., 2016	https://git.metabarcoding.org/obitools/obitools/wikis/home
Barque	Pipeline	--	https://github.com/enormandeau/barque
QIIME2	Pipeline	Bolyen et al., 2018	https://docs.qiime2.org
VSEARCH	M, Dr, QF, E, T†	Rognes et al., 2016	https://github.com/torognes/VSEARCH
Pear	M	Zhang et al., 2014	http://www.exelixis-lab.org/web/software/pear
FLASh	M	Magoč & Salzberg, 2011	https://sourceforge.net/p/flashpage
CASPER	M	Kwon et al., 2014	http://best.snu.ac.kr/casper/
Fastq-join	M	Aronesty, 2013	https://github.com/brwnj/fastq-join
Fastp	M, QF	Chen et al., 2018	https://github.com/OpenGene/fastp
Cutadapt	Dm, QF	Martin., 1994	https://github.com/marcelm/cutadapt https://cutadapt.readthedocs.io
Prinseq	QF	Schmieder & Edwards, 2011	http://prinseq.sourceforge.net
Flexbar	QF	Dodt et al., 2012	https://github.com/seqan/flexbar/wiki
Swarm	E	Mahé et al., 2015	https://github.com/torognes/swarm
Sintax	T	Edgar, 2016	https://www.drive5.com/usearch/manual/cmd_sintax.html

[†] Step abbreviations: M: merging, Dm: demultiplexing, Dr: dereplication, QF: quality filtering, E: PCR/sequencing error removal, T: taxonomic assignment

Table 2. Description of the six analyses steps, their objectives, and the parameters set for each programs compared

Analysis step	Objective	Program	Parameters
Merging reads	Assemble forward and reverse reads. Min. overlap =10 Max. overlap = 150 (Max mismatch = 25%)	illuminapairedend	--
		VSEARCH	--fastq_mergepairs –threads 1 –fastq_maxdiffpct 25
		Flash	-m 10 –M 150 –x 0.25 –t 1
		Fastq-join	-m 10 –p 25
		CASPER	-w 10 –g 0.25 –t 1 -j
		Pear	-v 10 –c 0 –n 0 –j 1
		Fastp	--merge –overlap_len_require 10 --overlap_diff_limit 15 –w 1 --overlap_diff_limit_percent 25
Demultiplexing	Assign reads to sample and remove primers. 0 mismatch on tags, max. 2 mismatches on primers	ngsfilter	-e 2
		Cutadapt	-g –j 1 –e 0 (or 0.12) –O 8 (or 15) --revcomp
Dereplication	Gather strictly identical sequences and keep count of reads abundance.	obiuniq	-m sample
		VSEARCH	--derep_fulllength –sizeout –fasta_width 0 –threads 1 --minseqlength 1
Quality filtering	Filter sequences shorter than 20bp and/or containing ambiguous bases	obigrep	-s’[ATCG]+$’–l 20
		VSEARCH	--fastx_filter –fastq_maxn 0 –fastq_minlength 20 –threads 1
		Cutadapt	-m 20 –max_n 0 –j 1
		Flexbar	--max-uncalled 0 –n 1 –min-read-length 20
		Prinseq	-min_length 20 –ns_max_n 0 -noniupac
		Fastp	-n 0 –l 20 –w 1
Error removal	Identify and remove PCR and sequencing errors (by clustering with proportion	obiclean	-r 0.05 –H
		VSEARCH	--cluster_unoise --sizein –minsize 1 –sizeout –threads 1

	of variants/parents)		–unoise_alpha 2 --minseqlength 20
		Swarm	-z –f –t 1
Taxonomic assignment	Assign reads to a species, with a 98% similarity threshold with the best match	ecotag	–m 0.98
		VSEARCH	--usearch_global –id 0.98 –fasta_width 0 –dbmask none --maxaccepts 20 --maxrejects 20 –blast6out –maxhits 20 --top_hits_only --qmask none --minseqlength 1 –threads 1 –dbmatched --matched
		Sintax	-sintax_cutoff 0.98 –threads 1

Table 3. Programs and parameters used in the pipelines studied

Pipeline	Step	Program used	Parameters
QIIME2	Demultiplexing	demux emp-paired	--p-golay-error-correction FALSE
	Primer removal	cutadapt trim-paired	--p-error-rate 0.12 --p-overlap 16
	Filtering and denoising	Dada2 denoise-paired	--p-trunc-len-f 0 --p-trunc-len-r 0 --p-trunc-left-f 0 --p-trunc-left-r 0 --p-max-ee-f 2 –p-max-ee-r 2 --p-trunc-q 2 --p-chimera-method none
	OTU calling	dbotu-q2 call-otus	--p-gen-crit 0.1 --p-abund-crit 0 --p-pval-crit 0.005
	Taxonomic assignment	Feature-classifier classify-sklearn	--p-confidence 0.7
Barque	Filter and trim raw reads	Trimmomatic	Min_hit_length 16 Crop_length 80
	Merge paired-end reads	Flash	-t 1 –z –m 20 –M 280
	Split amplicon	Python script split_amplicons_one_file.py	Max_primer_diff 8
	Taxonomic assignment	VSEARCH –usearch_global	--qmask none –dbmask none –id 0.98 –maxaccepts 20 –maxrejects 20 –maxhits 20 –minseqlength 20 –query_cov 0.6 –fasta_width 0

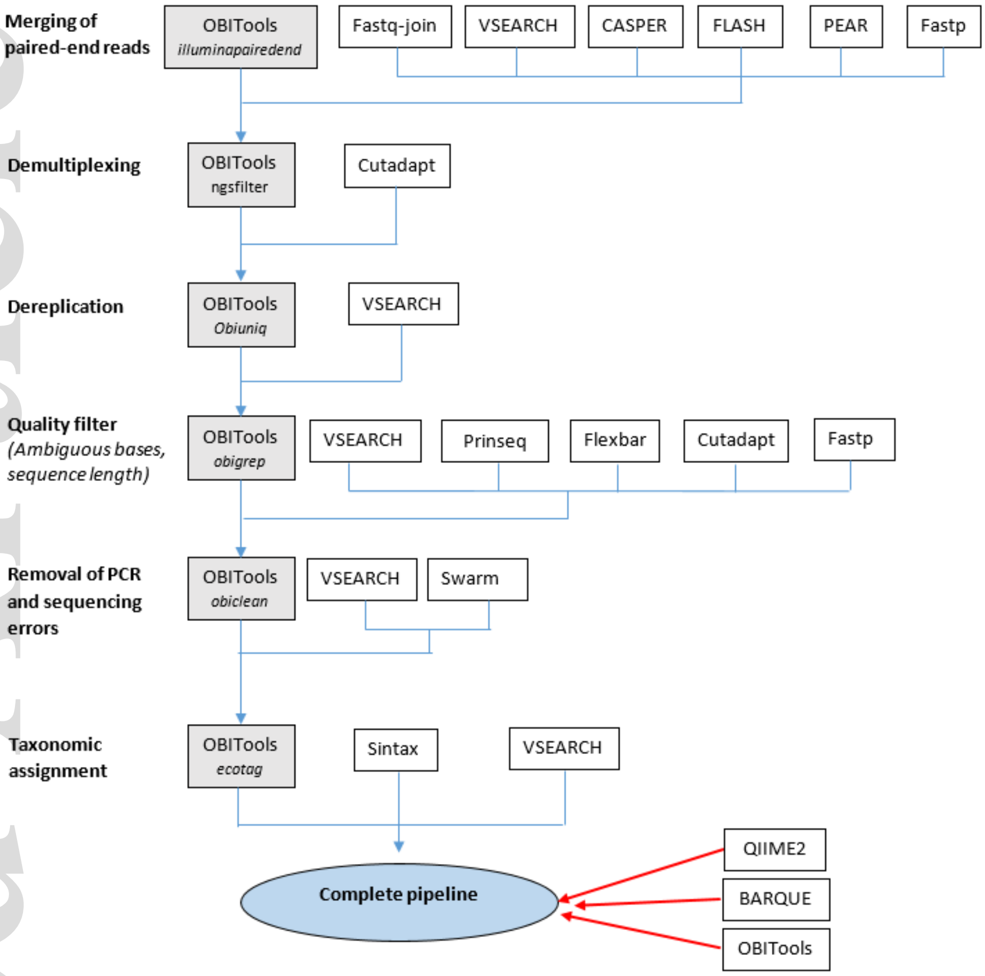

Figure 1. Protocol of the step-by-step program comparisons. The comparison of each program (white boxes) is arranged sequentially according to the suite of steps in the OBITools pipeline (grey boxes).

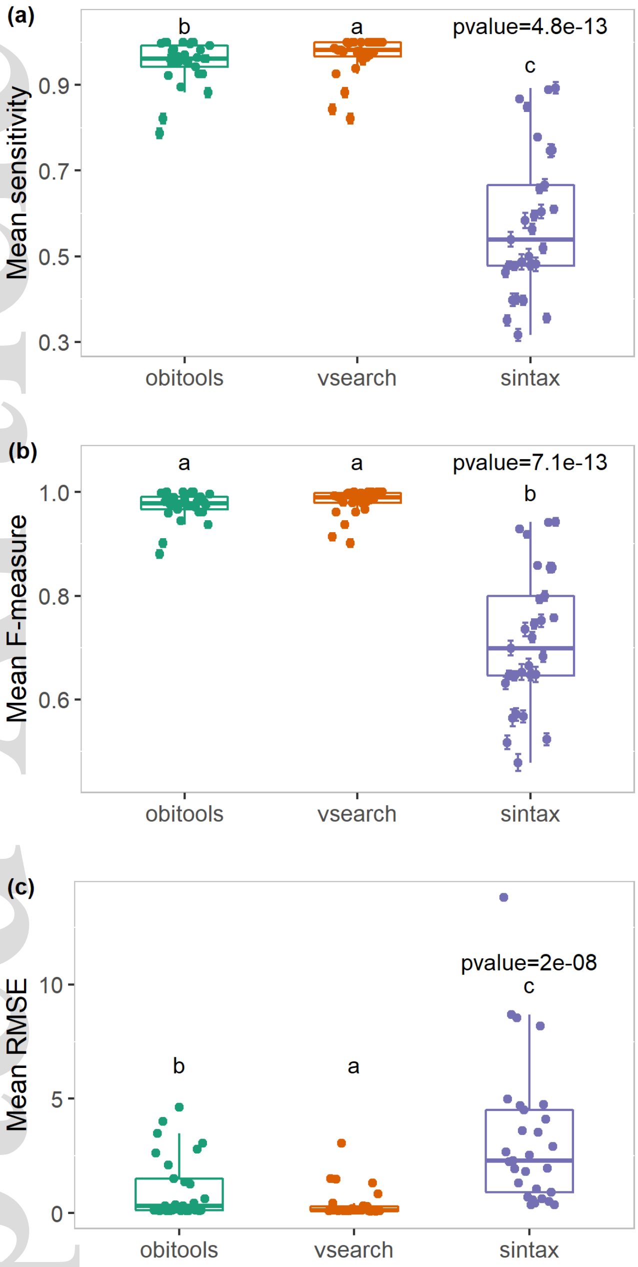

Figure 2. Compared performance indices of each program tested on the simulated dataset, for the taxonomic assignment step. The dots represent the mean index for the 12 replicates of each sample, with the standard error; the boxplot represents the median of the performance index and the first and third quartiles for the 29 samples. **(a):** Sensitivity calculated on the raw outputs of each pipeline, **(b):** F-measure calculated on the raw outputs of each pipeline, **(c):** RMSE calculated between observed abundances and expected abundances for each replicate of each sample. Letters indicate significant differences.

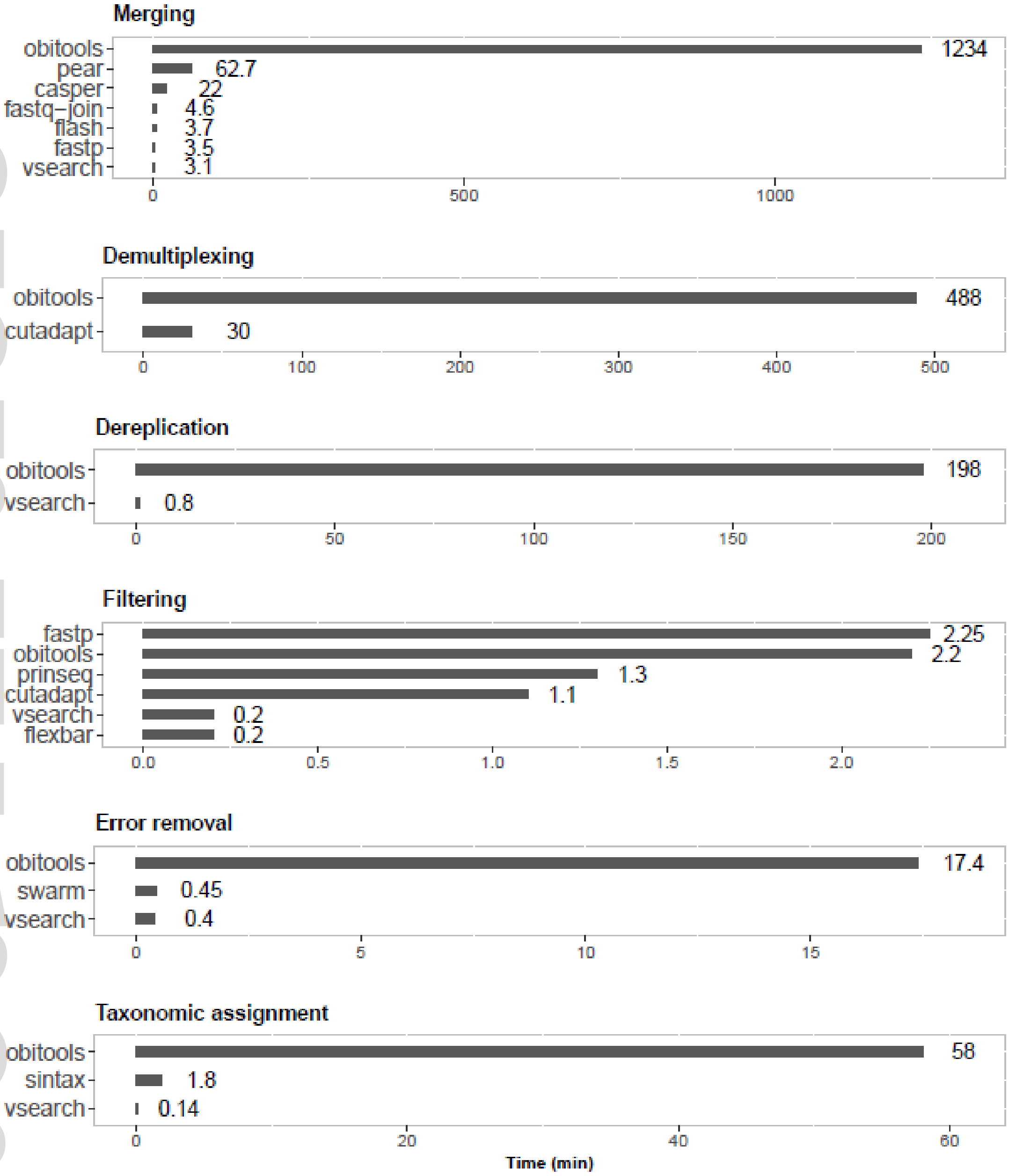

Figure 3. Execution time in minutes of each program tested for each step on the simulated dataset. Programs compared for the assembly, demultiplexing, dereplication, filtering, error removal, and assignment steps.

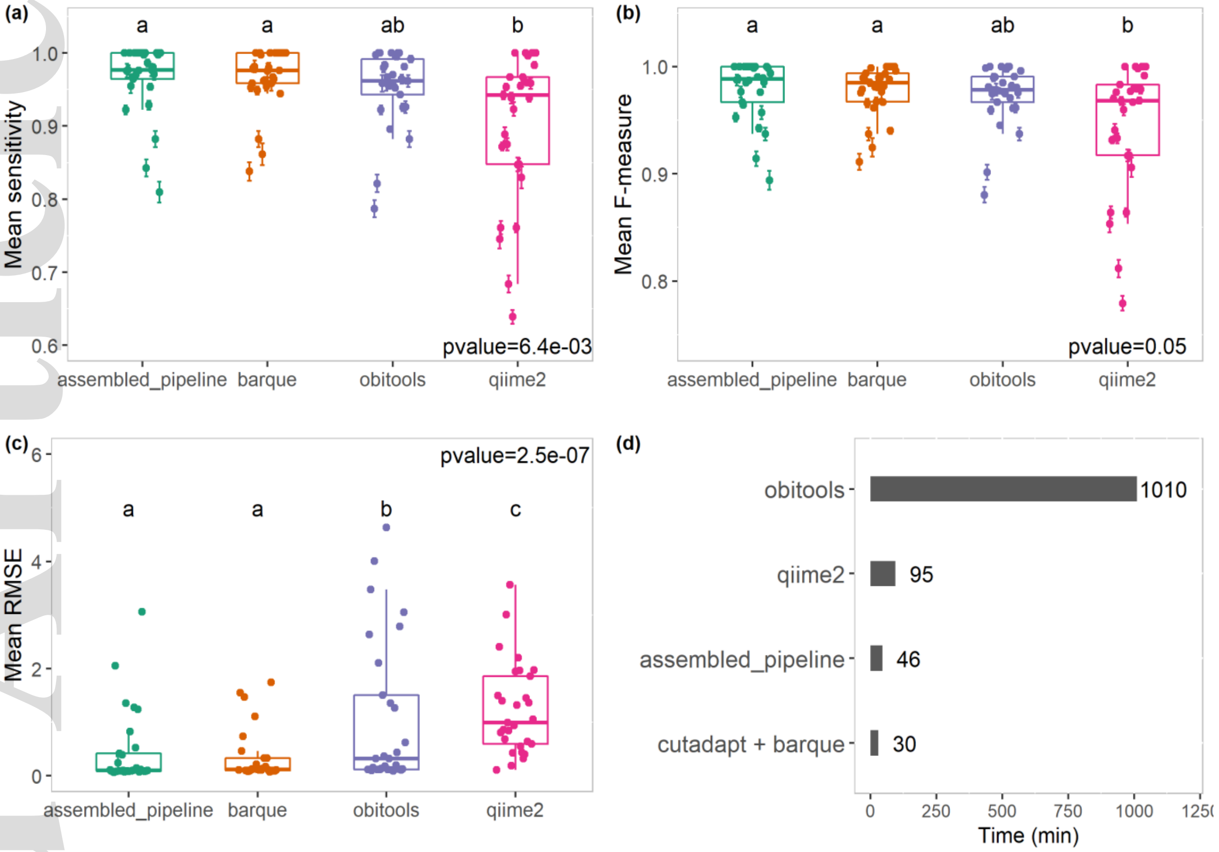

Figure 4. Performance indices of each pipeline on the simulated dataset. The dots represent the mean index for the 12 replicates of each sample, with the standard error; the boxplots represent the median of the index and the first and third quartiles for the 29 samples. **(a)** Sensitivity calculated on the raw outputs of each pipeline, **(b)** F-measure calculated on the raw outputs of each pipeline, **(c)** RMSE calculated between observed abundances and expected abundances for each replica of each sample, **(d)** Execution time of each pipeline.

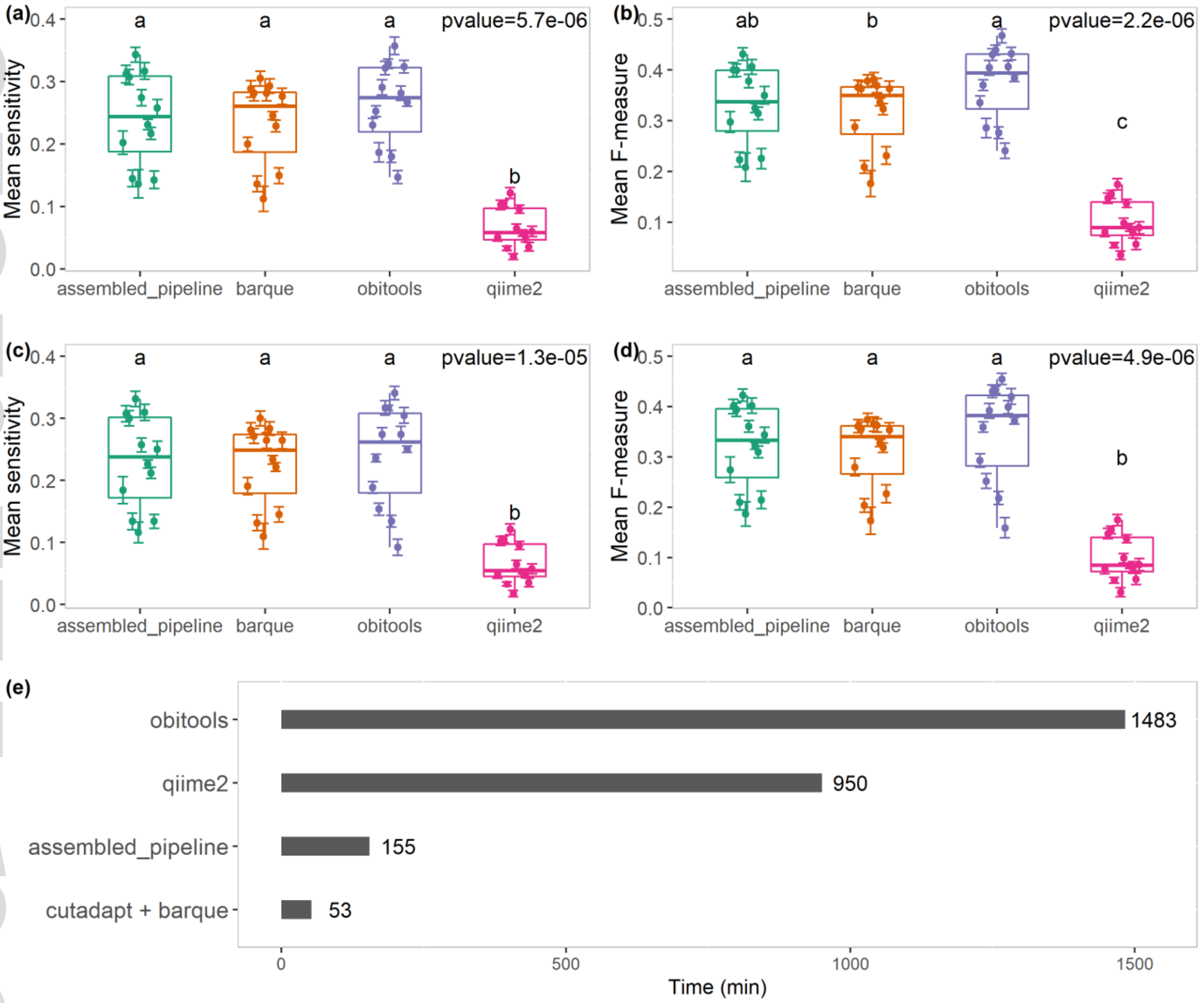

Figure 5. Performance index of the 4 pipelines on the real dataset. (a) Mean sensitivity, **(b)** Mean F-measure, **(c)** Mean sensitivity, after removing singletons from the dataset, **(d)** Mean F-measure, after removing singletons from the dataset, **(e)** Execution time of each pipeline.