MSc Bioinformatics
SCHOOL OF MEDICINE
UNIVERSITY OF CRETE

DISSERTATION

‘eDNA metabarcoding for biodiversity assessment: algorithm design and bioinformatics analysis pipeline implementation’

Zafeiropoulos Haris

Primary advisor: Evangelos Pafilis (IMBBC, HCMR)

Thesis Committee members:

Ioannis Tsamardinos (Computer Science Department, UoC)

Christos Arvanitidis (IMBBC, HCMR)

Pantelis Topalis (IMBB, FORTH)

Heraklion, 2018

ΠΜΣ Βιοπληροφορικής
ΙΑΤΡΙΚΗ ΣΧΟΛΗ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ

ΔΙΑΤΡΙΒΗ

‘μετακωδικοποίηση eDNA για την αξιολόγηση της βιοποικιλότητας: σχεδιασμός αλγορίθμων και υλοποίηση γραμμών εργασιών βιοπληροφορικής’

Ζαφειρόπουλος Χάρης

Κύριος επιβλέπων: Ευάγγελος Παφίλης

Τριμελής επιτροπή:

Ιωάννης Τσαμαρδίνος (Τμήμα Πληροφορικής, Πανεπιστήμιο Κρήτης)

Χρίστος Αρβανιτίδης (ΙΘΑΒΒΥΚ, ΕΛ.ΚΕ.Θ.Ε)

Παντελής Τοπάλης (ΙΜΒΒ, ΙΤΕ)

Ηράκλειο, 2018

Περίληψη

Το περιβαλλοντικό DNA (eDNA), δηλαδή γενετικό υλικό που έχει αποκτηθεί απευθείας από κάποιο περιβάλλον (έδαφος, ίζημα, νερό κ.λπ.) χωρίς κάποια εμφανή σημάδια βιολογικών πηγών από το οποίο προέρχεται [1],και η τεχνική της μετακωδικοποίησης, μια μέθοδος κωδικοποίησης DNA που επιτρέπει τη ταυτοποίηση μιγμάτων οργανισμών κάνοντας χρήση καθολικών εκκινητών της Αλυσιδωτής Αντίδρασης Πολυμεράσης (PCR), επιδιώκουν να γυρίσουν σελίδα στο τρόπο με τον οποίο η βιοποικιλότητα γίνεται αντιληπτή και παρακολουθείται. Ο συνδυασμός τους θεωρείται πως είναι μια γρήγορη μέθοδος αξιολόγησης της βιοποικιλότητας. Επιπλέον, η μετακωδικοποίηση του περιβαλλοντικού DNA είναι μια ολιστική προσέγγιση που, μόλις τυποποιηθεί, επιτρέπει μεγαλύτερη ικανότητα ανίχνευσης και σε χαμηλότερο κόστος σε σύγκριση με τις συμβατικές μεθόδους εκτίμησης της βιοποικιλότητας.

Παρόλο που η μετακωδικοποίηση κερδίζει έδαφος ως μια γρήγορη μη επεμβατική τεχνική εκτίμησης της βιοποικιλότητας, πρέπει να αντιμετωπιστούν πολλά θέματα της μεθόδου. Η μη ύπαρξη ενός τυποποιημένου πρωτοκόλλου για αυτό το είδος ανάλυσης, η μεροληψία συγγένειας του εκκινητή λόγω της χρήσης των γενικών εκκινητών και της διαφορετικής σχετικής πυκνότητας κάθε ακολουθίας στο δείγμα, καθώς και η δυσκολία προσδιορισμού των αφθονιών των ειδών, αποτελούν εμπόδια ζωτικής σημασίας για τις μελέτες βιοποικιλότητας.

Το κύριο μέλημα της παρούσας διατριβής, είναι η αντιμετώπιση ορισμένων από αυτά τα μειονεκτήματα σχετικά με την μετακωδικοποίηση του περιβαλλοντικού DNA, αφενός σχεδιάζοντας εκκινητές παρεμπόδισης για την αντιμετώπιση του "προβλήματος των εκκινητών" (ως τέτοιο, ορίζονται οι μεροληψίες που οφείλονται στη χρήση γενικών εκκινητών) στην περίπτωση των Μυκήτων, αφετέρου φτιάχνοντας μια τυποποιημένη γραμμή εργασίας για την ανάλυσή του.

Λόγω μεροληψιών που συμβαίνουν κατά την Αλυσιδωτή Αντίδραση Πολυμεράσης, ακολουθίες χαμηλής αφθονίας στα δείγματα σε σύγκριση με άλλες Ταξινομικές Λειτουργικές Μονάδες (OTUs), δεν ενισχύονται αποτελεσματικά. Οι Μύκητες συνήθως επικρατούν στα περιβαλλοντικά δείγματα και, συνεπώς, είναι υπεύθυνοι για ένα σημαντικό και ανεπιθύμητο θόρυβο στο προϊόν της Αλυσιδωτής Αντίδρασης Πολυμεράσης. Προκειμένου να ξεπεραστεί αυτό το πρόβλημα, δηλαδή για να αποφευχθεί η ενίσχυση των Μυκήτων κατά την Αλυσιδωτή Αντίδραση Πολυμεράσης, εκκινητές παρεμπόδισης για τα δύο γονίδια δείκτες (16S rRNA και COI) σχεδιάστηκαν *in silico*. Οι προβλεπόμενοι εκκινητές παρεμπόδισης αξιολογήθηκαν, επίσης *in silico*; στην περίπτωση γονιδίου δείκτη 16S τα αποτελέσματα ήταν πολλά υποσχόμενα μιας και το ζεύγος των εκκινητών παρεμπόδισης που προέκυψε, δεν εμπόδισε την ενίσχυση των Βακτηρίων. Στην περίπτωση του γονιδίου δείκτη COI, τα αποτελέσματα δείχνουν ότι υπάρχει ένα μικρό ποσοστό ευκαρυωτικών αλληλουχιών που αποκλείονται, μαζί με την επιθυμητή απόφραξη των μυκητιακών ακολουθιών. Ωστόσο, τα σχεδιασμένα ζεύγη εκκινητών παρεμπόδισης για αμφότερα τα γονίδια δείκτες, φαίνεται πως έχουν τη δυνατότητα να ενεργούν ως τέτοιοι και πλέον θα πρέπει να δοκιμαστούν περαιτέρω στο εργαστήριο.

Επιπλέον, καμία τυποποιημένη ροή εργασίας για την ανάλυση των εκατομμυρίων αλληλουχιών που προκύπτουν ως προϊόν της Αλυσιδωτής Αντίδρασης Πολυμεράσης ανά πείραμα, έχει αναπτυχθεί. Πληθώρα εργαλείων βιοπληροφορικής παρέχονται για κάθε βήμα της ανάλυσης αυτής, ωστόσο δεν υπάρχει μια συγκεκριμένη συλλογή εργαλείων που να έχει αξιολογηθεί *a priori* και να έχει δοκιμαστεί επαρκώς, ώστε να μπορεί να χρησιμοποιηθεί ως το "χρυσό πρότυπο" για κάθε ανάλυση μετακωδικοποίησης.

Για το σκοπό αυτό, ο στόχος αυτής της μελέτης ήταν να οικοδομηθεί μια πλήρης και αποτελεσματική ροή εργασίας (με τον τίτλο "P.E.M.A.") για αμφότερα τα γονίδια δείκτες 16S και COI. Μια γλώσσα προγραμματισμού ειδική για τη σχεδίαση ροών εργασίας που σχετίζονται με την επεξεργασία δεδομένων, που ονομάζεται Big Data Script (BDS), χρησιμοποιήθηκε για το σχεδιασμό του P.E.M.A., του οποίου τα αρχεία εισόδου είναι ακατέργαστα αρχεία ανάγνωσης αλληλουχιών (.fastq αρχεία). Διαφορετικοί αλγόριθμοι ομαδοποίησης (Μοριακών) Λειτουργικών Ταξινομικών Μονάδων ((M) OTU) καθώς και διαφορετικές μέθοδοι απόδοσης ταξινομίας σε αυτές, παρέχονται στον χρήστη, ανάλογα με το επιλεγμένο γονίδιο δείκτη και τις ιδιαιτερότητες του συνόλου των δεδομένων του πειράματός. Το P.E.M.A. αξιολογήθηκε χρησιμοποιώντας δύο σύνολα δεδομένων από δημοσιευμένες μελέτες και τα παραγόμενα αποτελέσματα ήταν παρόμοια με εκείνα των μελετών. Προτείνεται ότι το P.E.M.A. μπορεί να χρησιμοποιηθεί για την ακριβή ανάλυση μελετών μετακωδικοποίησης περιβαλλοντικού DNA και συνεπώς μπορεί να ενισχύσει την εφαρμογή της βιοποικιλότητας επόμενης γενιάς σε μελέτες αξιολόγησης.

eDNA metabarcoding for biodiversity assessment: algorithm design and bionformatics analysis pipeline implementation

Zafeiropoulos Haris

November 29, 2018

Abstract

Environmental DNA (eDNA), i.e. genetic material obtained directly from environmental samples (soil, sediment, water, etc.) without any obvious signs of biological source material [1], and metabarcoding, a DNA barcoding method that allows the identification of a mixture of organisms using universal PCR primers, attempt to turn the page into the way biodiversity is perceived and monitored. Their combination is considered to be a rapid method of biodiversity assessment. Furthermore, eDNA metabarcoding is a holistic approach that, once standardized, allows for higher detection capacity and at a lower cost compared to conventional methods of biodiversity assessment.

Even though metabarcoding is gaining ground as a fast non-invasive biodiversity assessment technique, numerous issues of the method need to be addressed. The nonexistence of a standardized protocol for this kind of analysis, the primer affinity bias due to the use of universal primers and the different relative density of each sequence in the sample, as well as the difficulty in determining species abundances, are all hurdles of crucial importance for biodiversity studies.

This MSc thesis' main focus is the troubleshooting of some of those drawbacks on eDNA metabarcoding, by designing blocking primers to address "the primer issue" (biases due to the use of generic primers, are defined as such) in the case of Fungi and by building a standardized bioinformatic pipeline for its analysis.

Due to PCR biases, sequences of low abundance in the samples compared to sequences of other OTUs, are not amplified efficiently. Fungi usually prevail in environmental samples and, thus, they are responsible for a considerable and undesirable noise in the PCR product. In order to overcome this problem, i.e. in order to prevent PCR amplification of Fungi, blocking primers for the two marker genes (16S rRNA and COI) were designed *in silico*. The predicted blocking primers were evaluated, also *in silico*; in the case of 16S marker gene the results were promising as the blocking primer pair did not prevent amplification of Bacteria. In the case of COI marker gene, the results show that there is a small percentage of eukaryotic sequences that are blocked, along with the desired blockage of fungal sequences. However, the designed blocking

primer pairs for both marker genes have the potential to act as such and should be further tested in the laboratory.

Moreover, no standardised pipeline for the analysis of the millions of the amplicon reads per experiment has been developed. Numerous tools for each step of the analysis are provided, but there is no set of tools which are *a priori* evaluated and benchmarked and thus can be used as the "golden standard" of each metabarcoding analysis. To this end, the goal of this study was to build a complete and efficient pipeline (entitled "P.E.M.A.") for both 16S and COI marker genes. A programming language for data processing pipelines, called Big Data Script (BDS), was used for the design of P.E.M.A. whose input are raw sequence read files (.fastq format). Different (Molecular) Operational Taxonomic Unit ((M)OTU) clustering algorithms and taxonomic assignment approaches are provided for the user to choose, depending on the chosen marker gene and the particularities of the dataset. P.E.M.A. was evaluated using two datasets from published studies and the produced results were similar to those of the studies. It is suggested that P.E.M.A. can be used for accurate eDNA metabarcoding analysis and, hence, it can enhance the applicability of next-generation biodiversity assessment studies.

Contents

Part I

Main introduction

1 Biodiversity and biodiversity monitoring

Biodiversity is a basic concept and challenge in biology. "Biological diversity" means the variability among living organisms from all sources including, inter alia, terrestrial, marine and other aquatic ecosystems and the ecological complexes of which they are part; this includes diversity within species, between species and of ecosystems" (CBD). Traditoinally, biodiversity is recognized at three levels of the biological organization: genes, species and ecosystems (habitats). Biodiversity is also subjected to the basic process of change, that is evolution. This alone provides the biological systems with the opportunity to adapt in a constantly changing environment. Biodiversity assessment is key to understanding the relationship between biodiversity and ecosystem functioning (BEF) [2]. The lack of knowledge on the state of biodiversity – especially since the majority of species on Earth have yet to be described [3] – is a serious impediment. The ever increasing monitoring of species and populations aims at addressing this issue and, in a stepwise fashion, produce reliable distribution patterns and population size estimates.

Traditionally, biodiversity monitoring has been based on the taxonomic identification of the physical specimens of the species. Such approaches are valuable as they provide demographic information, such as sex, maturity, age, and abundance [4] and are necessary to expand further the developed species databases. However, they are often limited by low identification rates, high field costs, time-intensive sampling and their potentially ecological destructive nature [5]. Hence, alternative and efficient techniques for large - scale biodiversity monitoring could be a huge support [1]. Genetic-based techniques like DNA barcoding and meta-barcoding, attempt to play a complementary role to traditional ones [6]. Furthermore, they offer the potential of looking into the past and how biodiversity and ecosystems used to be, as well as the prediction of what could happen in the future in a specific ecosystem is undeniably a remarkable task.

Main focus of this thesis is on eDNA metabarcoding analysis, the troubleshooting of this technique and the making of a standardized bioinformatic pipeline for it.

2 Metabarcoding and Environmental DNA

Metabarcoding is a relatively recent approach that focuses on the simultaneous detection of a large number of species [7]. Its main principles and characteristics as well as the most significant pros and cons, are mentioned below.

Metabarcoding is a method of DNA barcoding that uses universal PCR primers to

identify DNA from a mixture of organisms [8]. Linguistically the term "metabarcoding" can be interpreted in multiple ways leading sometimes to misunderstandings and confusion. "Meta" comes from the greek word for "after" and easily one, for example, might erroneously think about something that comes after the DNA-level. Despite this, in this thesis the term metabarcoding will still be used due to its strong prevalence and community recognition. It should be stated, however, our belief is that the term "environmental genomics" would be more appropriate.

2.1 Metabarcoding

Metabarcoding is a novel method that builds on the added value obtained once DNA barcoding is coupled with Next Generation Sequencing (NGS) [9].

DNA barcoding is a taxonomic method that uses a short genetic marker on an Operational Taxonomic Unit's (OTU's) DNA in order to identify it as belonging to a particular taxon, using a pre-existing classification. The DNA sequence of the selected region used for recognition is called, barcode. DNA barcoding relies on a small piece of the genome found in a broad range of species and usually located on the mitochondrial genome for animals or the chloroplast for plants [10].

The success of DNA barcoding relies on the coexistence of two factors: 1. a genetic marker universally present in every species, which could be easily sequenced using standardized protocols. This marker should have enough sequence variability to allow distinction among related OTUs (e.g. species) but must be surrounded by regions conserved enough so that universal primers could be designed. 2. a comprehensive public database containing the known sequences of this marker for the maximum possible number of different OTUs. Such a database should be searchable by sequence alignment algorithms, so that experimentally derived sequences could be matched to known species [11]. OTU identification via DNA-barcoding is evidently as good and reliable as complete and accurate this reference database is.

Metabarcoding, in comparison to barcoding, uses universal PCR primers to mass-amplify a taxonomically informative gene from mass collections of organisms or from Environmental DNA (eDNA). The prefix "meta" is added when multiple species are identified from a single sample. Metabarcoding, enables simultaneous high throughput multi-taxa identification by using the extracellular and/or total DNA extracted from complex samples containing DNA of different origins [12]. In this case, the PCR product is loaded to a high-throughput sequencer, and the output is a long list of DNA sequences. Such sequence collections must be verified by fieldwork and traditional taxonomic identification to confirm the presence or absence of particular species. These metabarcode data sets are taxonomically more comprehensive, many times quicker to produce, and less reliant on taxonomic expertise [13].

2.2 Environmental DNA

Metabarcoding is a technique applied, among others, on Environmental DNA (eDNA) samples. With the term eDNA we refer to DNA that can be extracted from environmental samples (such as soil, water or air) without any taxon targeted isolation first [14]. As a matter of fact, eDNA is DNA that has been released by an organism into the environment. Such free DNA molecules are present everywhere and can be deposited through skin flakes, urine, faeces, eggshells, hair, saliva, insect exuviae, regurgitation pellets, feathers, leaves, root cap cells, in rare cases pollen, or in living prokaryotes through the secretion of plasmid and chromosomal DNA [15]. Total eDNA contains both cellular DNA (living cells or organisms) and extracellular DNA (resulting from natural cellular death and subsequent destruction of cellular structure).

A key feature of marine samples is they contain limited concentrations of extracellular DNA. The main reason for such limited concentration of extracellular DNA, is degradation. In different environments, DNA degradation occurs in different rates. When it comes to marine samples, it has its maximum effect due to the presence of water [16] [17]. Thus, DNA lasts only a few weeks, contrary to soil samples, where even ancient DNA can be found [18]. Besides water, other factors causing DNA hydrolysis (and hence influence the chance of detecting a species) are endonucleases, UV radiation, Bacteria and Fungi.

The above is particularly true for marine water column samples. Marine sediment samples, however, are characterised by extracellular DNA turnover that is about 200 times slower. The pool size of extracellular DNA in marine sediments is the result of complex interactions, including DNA inputs from the photic layer through particle sedimentation, autochthonous DNA production, and degradation or utilization or both by heterotrophic organisms [19].

Put together, eDNA and metabarcoding are a powerful tool, called the eDNA method. Species that are passing by a specific location, even species that might used to reside in a place and are now extinct (mostly from soil samples) can be detected. So, even population dynamics of the species of an ecosystem can be explored, too.

3 NGS and Illumina Mi-Seq: a brief overview

The NGS technology is well-known and used by numerous researchers [20]. Several NGS sequencers have been developed and used since the advent of pyrosequencing [21], with those manufactured by Illumina being rendered as the most popular choice. However, the way this technology works is not commonly known. As a result, a lot of analyses are condemned to errors due to wrong usage of the Illumina output and a series of metabarcoding disadvantages are caused by the way NGS works itself.

As it is shown in the figure below (Figure 1), Illumina has a special flow cell in which the adaptors are attached to. In paired end experiments, there are two adapter se-

quences. At first, the read_1 sequencing with 5' -> 3' direction, starts exactly after the reverse complement of adaptor_2 and finishes after the number of cycles that the user has selected and that are specified in the chosen sequencing kit. This way, in case that the amplicon is shorter than the number of cycles, there is the risk to sequence even parts of the adaptor_1. So, the correct choice of sequencing kit depends on the length of the reads. When read_1 is completed, it is removed, the initial sequence folds and the reverse complement of adaptor_2 binds with adaptor_2. Due to the presence of DNA polymerase, the sequence that starts with adaptor_2 starts to extend. Finally, the initial sequence is removed and the read_2 - reverse complement of read_1 - starts to be sequenced. Lastly, two .fastq output files are produced from the Illumina sequencer, one for read_1 and another for read_2. After pre-processing of these two, a new file is created, containing the final merged sequence.

The sequencing is possible because with each cycle fluorescent tagged nucleotides compete for addition to the growing chain. After the addition of each nucleotide, the clusters (subsequent solid-phase bridge amplification generates up to 1000 copies in close proximity) are excited by a light source and a characteristic fluorescent signal is emitted. The sequencing-by-synthesis technology uses fluorescently labelled reversible terminator-bound dNTPs (deoxyribonucleotides: A,C,G,T) for the polymerization. Only one base is added in each step due to the 3′ termination of the incorporated nucleotide. The fluorophores are illuminated by a red laser for A and C and a green laser for G and T and imaged through different filters to identify the four different nucleotides. The fluorescent labels and the 3′ terminators are then removed in order for the next cycle to commence. [22].

The number of cycles determines the final length of the read. It is characteristic of NGS technology that as the sequencing length increases, the quality of the sequencing decreases, i.e. sequencing errors are unevenly distributed along the length of the sequences [22]. As a result, paired-end sequencing returns considerably improved results, as the part of the amplicon that is of low quality in read_1, is of high quality in read_2, and vice-versa.

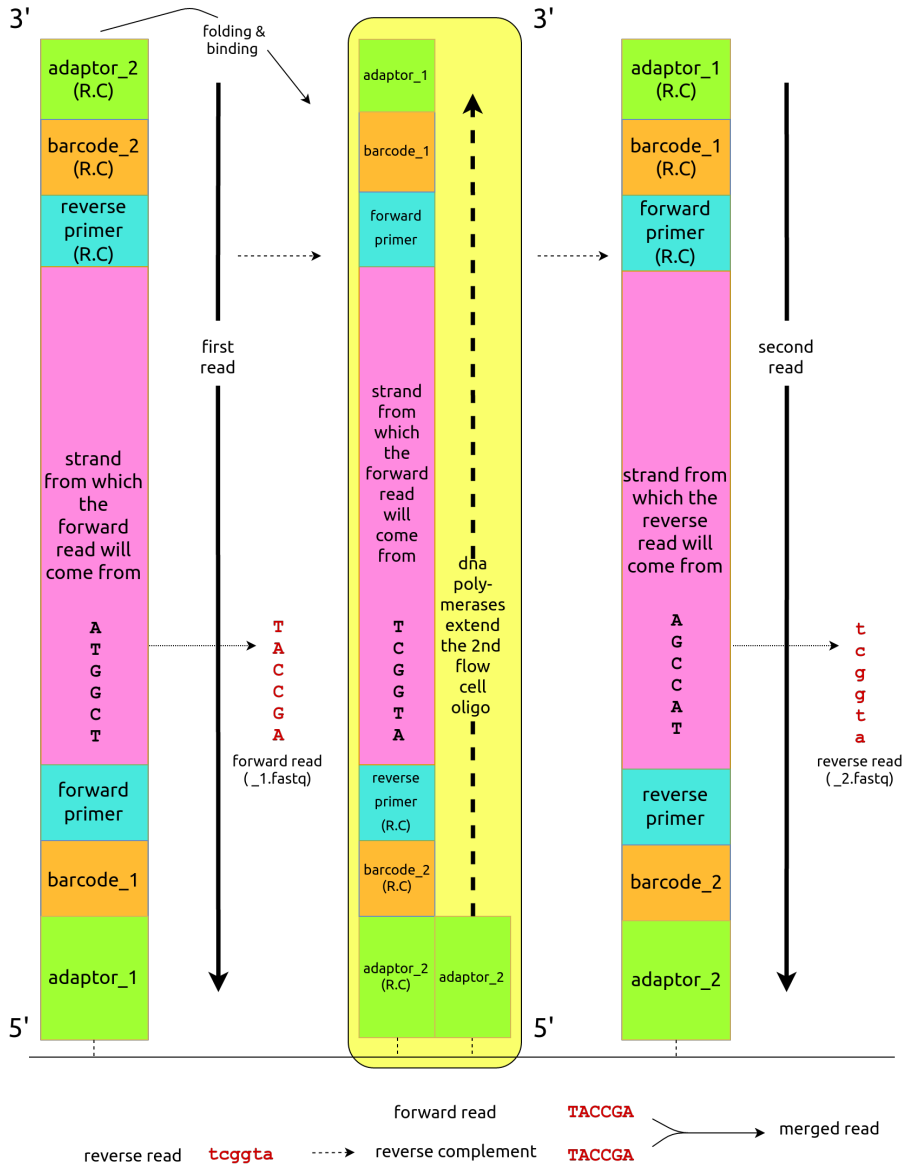


Figure 1: Technology of Illumina sequencing for the case of paired-end sequencing.

4 Advantages and disadvantages of eDNA metabarcoding

The eDNA method is a powerful tool, both for biodiversity monitoring and for exploration of the evolutionary processes.

The eDNA method overcomes weaknesses present in traditional techniques, like: 1) incorrect identification of cryptic species or juvenile life stages, 2) the continuous decline in taxonomic expertise, and 3) the invasive nature of some survey techniques that can sometimes cause problems in the ecosystem under study [1].

eDNA, along with NGS and metabarcoding, gives us the ability not just to identify the species present in a sample, but also to: 1) explore ecosystem-level processes, 2) acquire quantitative indices for species analysis and 3) examine community diversity and dynamics

For now, most of the attempts for abundance estimation, especially in the case of Eukaryoters, rely on the assumption that the amount of eDNA that is collected, is correlated to the abundance or standing biomass of each taxa that are contained on it .

The above are possible due to the eDNA method's pioneering sensitivity for detecting rare or difficult-to-sample taxa [23]. Overall, the eDNA method shows higher detection capability and cost - effectiveness compared to traditional methods [1].

On the other hand, the main disadvantage of the eDNA method is that until now the relationship between the amount of eDNA and the abundance of the target species has not been resolved. As a result, most of the times, mainly when Eukaryotes are under study, their absolute abundances cannot be determined and only relative abundance estimation is possible [24] and that .

In, mainly when Bacteria are under study addiis sotion, the eDNA method only retrieves information about the presence or absence of the target species/taxa and their relative abundance, to some extent. Information regarding factors such as the life stage, reproduction, fitness of a species cannot be derived. Also, hybrids cannot be distinguished from their maternal species, as most eDNA studies focus on mitochondrial DNA that is inherited only from the mother [24].

However, the method is gradually but significantly improving becasue the eDNA metabarcoding analysis gets standardized and as genetic markers, able to provide such information, are developed. Typical example of such marker genes are some age - related epigenetic markers, such as TET2, CDKN2A, HoxA9 [25]. More information on marker genes and their significance is presented in the next section.

For both the design of blocking primers and the eDNA metabarcoding analysis that were implemented, bioinformatic tools and a bioinformatic pipeline for the analysis of the sequencing were key to our study.

Therefore, the chosen approach included: (a) the *in silico* design of blocking primers in order to prevent PCR amplification of Fungi for both 16S and COI genes and (b) the set up of a bioinformatic metabarcoding pipeline in order to analyze the results of NGS.

Part II

Blocking primer design *in silico*: prevention of PCR amplification of fungal 16S rRNA and COI genes in metabarcoding analyses

1 Introduction

1.1 Marker genes and their significance in (meta-)barcoding

The eDNA metabarcoding is a very powerful method, allowing the detection of numerous species as well as taxa of higher taxonomic level, without any prior knowledge of species distribution in the study area. For such a universal approach, marker genes with: adequate taxonomic coverage to support the identification of groups of interest sufficient sequence divergence to resolve species, and indicate relative abundance of taxa present are needed [26]. Marker gene is an orthologous gene group which can be used to distinguish between taxonomic lineages.

To achieve adequate taxonomic resolution, marker genes should show little or no variability within species but adequate variability between species, and should be flanked by PCR primer - binding sites that are sufficiently conserved to minimize taxonomic bias.

As mentioned by Kim and Chun (2014), *"Ribosomal RNA genes have been used as standard phylogenetic markers in molecular taxonomic studies since the pioneering studies on the tree of life. Their ubiquitous distribution across all archaeal and bacterial lineages, evolutionarily conserved nature, and a wide range of variable regions facilitated the use of rRNA genes for a variety of taxonomic purposes"* [27].

A typical example for a gene with these characteristics is 16S rRNA in which both conserved and variant regions are found. As it is shown in Figure 2, the bacterial 16S gene contains nine hypervariable regions (V1-V9) ranging from about 30-100 base pairs long that are involved in the secondary structure of the small ribosomal subunit. Highly conserved sequences can be found between hypervariable regions, enabling the design of universal primers that can reliably amplify the same sections of the 16S sequence across different taxa.

While most of the early studies on Bacteria and Archaea focused on the 16S rRNA

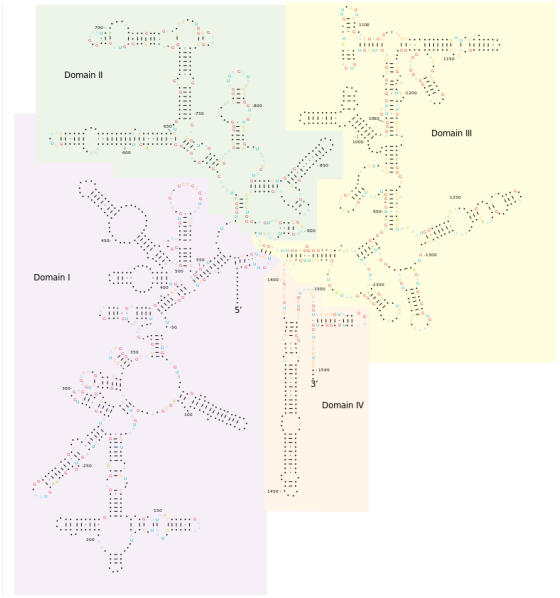


Figure 2: The structure of 16S rRNA gene and its 4 discrete domains.

gene, other taxonomic groups employed a diverse set of loci from the analogous eukaryotic rRNA gene array (e.g. ITS, 18S or 28S rRNA), chloroplast genes (for plants) and mitochondrial DNA (for multicellular animals) in an attempt for species - specific resolution. Both microfauna and meiofauna present exceptional results when Cytochrome c oxidase subunit I (COI) is used as marker gene [28].

The best case scenario for metabarcoding experiments, would be the existence of a single marker gene that could identify each and every species from every possible kingdom. However, a marker gene like that has not been found yet, and may never be. Hence, traditional universal markers from DNA barcoding are used. These barcodes were originally detected by observing sequence alignments and locating a pair of conserved regions flanking a variable one [29].

1.2 The primer issue

Metabarcoding primers for the marker genes mentioned before, often target short regions (<300 base pairs, bp) of multi - copy DNA (mitochondrial or ribosomal) to facilitate the amplification of potentially degraded eDNA and to minimize taphonomic biases (i.e. the fact that not all organisms have the same chance of being preserved and thus, some may be missing from the fossil record). Due to the short length of PCR products, they are also useful in other scenarios where DNA is likely to be degraded, such as ancient DNA, diet analysis and forensic studies [30]. The length for a suitable metabarcoding marker should be small enough to be informative and, at the same time, amplifiable in degraded DNA samples; ideally, its length should not exceed 350 bp.

DNA metabarcoding' s main drawback, is that by using generic primers, primer affin-

ity bias leads to certain sequences (species) amplifying less efficiently than others. This will, in turn, limit the results to species with best primer affinity or to species, which are already known to be locally present so specific - primers can be designed. Due to this PCR bias, less concentrated sequences are often not amplified because PCR favours the dominant DNA types [31]. Hence, some species may skew a number of others completely. However, such limitations will tend to become less crucial due to optimization and publication of generic primers for metabarcoding studies [1].

In addition, it should be mentioned that primer efficiencies across different taxa greatly hinder species abundance assessments using PCR - based approaches. Thus, it is not possible to accurately estimate species biomass or even abundance in diverse environmental samples using amplification-based sequencing protocols. For accurate estimates of biomass, or even rough estimates, a PCR - free approach is needed which, however, requires further development [8].

The universality or specificity of the primer set depends on which is the main research goal. If only a specific group is targeted, then a primer set specific for this group could be used. If the whole community of eukaryotic organisms is targeted, then the designing of a primer set as universal as possible would be necessary. Furthermore, two attributes are crucial and must be kept in mind for the design of primers for metabarcoding analysis: fragment length and universality or specificity of the primers to be designed [32].

The potential power of DNA metabarcoding as it is currently implemented is mainly limited by its dependency on PCR and by the considerable investment needed to build comprehensive taxonomic reference libraries. However, the near-term future of DNA metabarcoding has an enormous potential to boost data acquisition in biodiversity research, especially considering the impressive progress in DNA sequencing [7].

1.3 Fungi: a major problem when other groups are studied

It is because of those PCR biases, that make in many cases, a whole taxonomic group or even a single species (e.g. in molecular diet studies, where predator DNA is inevitably present in great surplus of prey - derived DNA) to create serious problems in metabarcoding. In marine sediment samples, Fungi are such a group when microbes and macrofauna are studied. Marine Fungi are known to originate from a wide variety of habitats within the marine environment [33]. In addition, Fungi play major roles as symbionts, pathogens, or decomposers in natural and managed ecosystems [34]. As a result, when microbes and macrofauna are studied, it is possible that due to their abundance and to the "primer issue", they will create an unwelcome noise.

Fungi, as they are eukaryotic organisms, carry both COI and 16S rRNA genes, the latter of which is used for the assessment of microbial communities. So, in both cases of marker genes, if universal primers are used then the fungal sequences will be amplified too.

Hence, Fungi can create a significant "noise" in the PCR products, especially due to the large groups they form and their characteristic variety.

This noise should be avoided for the following reasons: (a) If a particular DNA molecule is present in sample but in a very small quantity, it is quite likely not to be amplified if there is a large amount of other molecules to which the universal primer will bind (b) The large amount of fungal DNA will also pose a problem in any attempt to quantify - estimate the of abundance.

In order to remove the aforementioned noise, one solution would be to exclude the sequences that come from fungal species at the sequencing processing step. However, this could not be the appropriate solution; marker genes of low abundant species would not get amplified during the library preparation and fungi would have already skewed them at this step. Also, next generation sequencers are rather sensitive when the loaded libraries are of variable concentration; hence, Fungi would have skewed other species even during the sequencing step.

Based on the above, the proper solution to remove the fungal associated noise was considered the *in silico* design, and subsequent usage, of relevant blocking primers .

1.4 Blocking primers: a way to minimize amplification in target groups

Blocking primers are modified primers which overlap with one versatile primer binding site and extend into group specific sequences. They prevent DNA amplification for this group of sequences but simultaneously enable amplification of DNA from all the other sequences, i.e. taxa [35]. Thus, with the usage of blocking primers it is possible either to avoid the amplification of a specific species (species - specific blocking primers) or even of more numerous groups of related taxa. The latter however, is a more challenging, non trivial task for both biological and computational reasons that will be explained in the relevant section.

They are synthesized like conventional amplification primers, but modified with the addition of a C3 spacer at the 3' end, resulting in full inhibition of enzymatic elongation of the primer. Blocking primers can either compete directly with the amplification primers (annealing inhibiting blocking primers) or prevent elongation by binding onto the fragment in between the two amplification primers (elongation arrest blocking primers) [36]. Hence, blocking primers are oligos that bind to the target group' s DNA by preference but are modified so that they do not prime amplification [31].

For the purpose of this research, blocking primers that target marine Fungi, were designed *in silico*, both for the case of 16S rRNA and COI, in order to assist the in vitro experiments and finally achieve more precise results from metabarcoding analysis for microbes and eukaryotic macro - organism communities.

1.5 What is all about? Bioinformatics in the service of biodiversity

Proper selection of oligonucleotide primer is critical for PCR, DNA sequencing, and oligo-hybridization. Bioinformatics tools and software that are used in the case of simple amplification primers, can be also used for the designing of proper PCR blocking primers. Various bioinformatics programs are available for designing primers from a set of template sequences. Such programs help in primer design , but wet lab validation is further required before proceeding in their synthesis [37].

2 Methods

2.1 Primer design and performance test

The aim was the design of sets of blocking primers for marine Fungi within the COI barcoding region and the V3–V4 region of the 16S rRNA gene. The pipeline that was followed in both cases is presented in the sections below, mentioning any differences among the implementation of it in each case.

2.1.1 Sequence retrieval

Both COI and 16S sequences for Fungi, were acquired from GenBank [38] using Entrez E – utilities [39]. In addition, in order to be able to evaluate our predicted blocking primers, a “reference” database for COI sequences from every organism that has been recorded had to be created.

2.1.2 Sequence filtering

Subsequently, sequences were filtered in order to remove those that were neither marine nor from the locus of interest. Additionally, sequences whose length would be a problem in the alignment step, e.g. whole genome entries, were also removed. A series of filtering steps were followed as shown in Figure 3, with no apparent differences between the datasets with the fungal sequences of the two marker genes.

More specifically, at first the headers of the sequences were isolated and specific terms were looked for in each of them. If any of these terms was found in the header, then the sequence in which the header corresponded to, was removed from our dataset. Such terms were words as “ITS”, “internal”, “soil”, “whole genome”, “complete genome” etc. For this task, several bash commands (e.g. "grep -vi") were used (see Appendix 1).

Hence, in this way sequences that derived either from whole genome sequencing or from sequencing of the entire gene that our target region is located, constituting just a small fraction of it can be removed. Also, sequences that are referring to soil Fungi and also this information is mentioned in their title, can be removed.

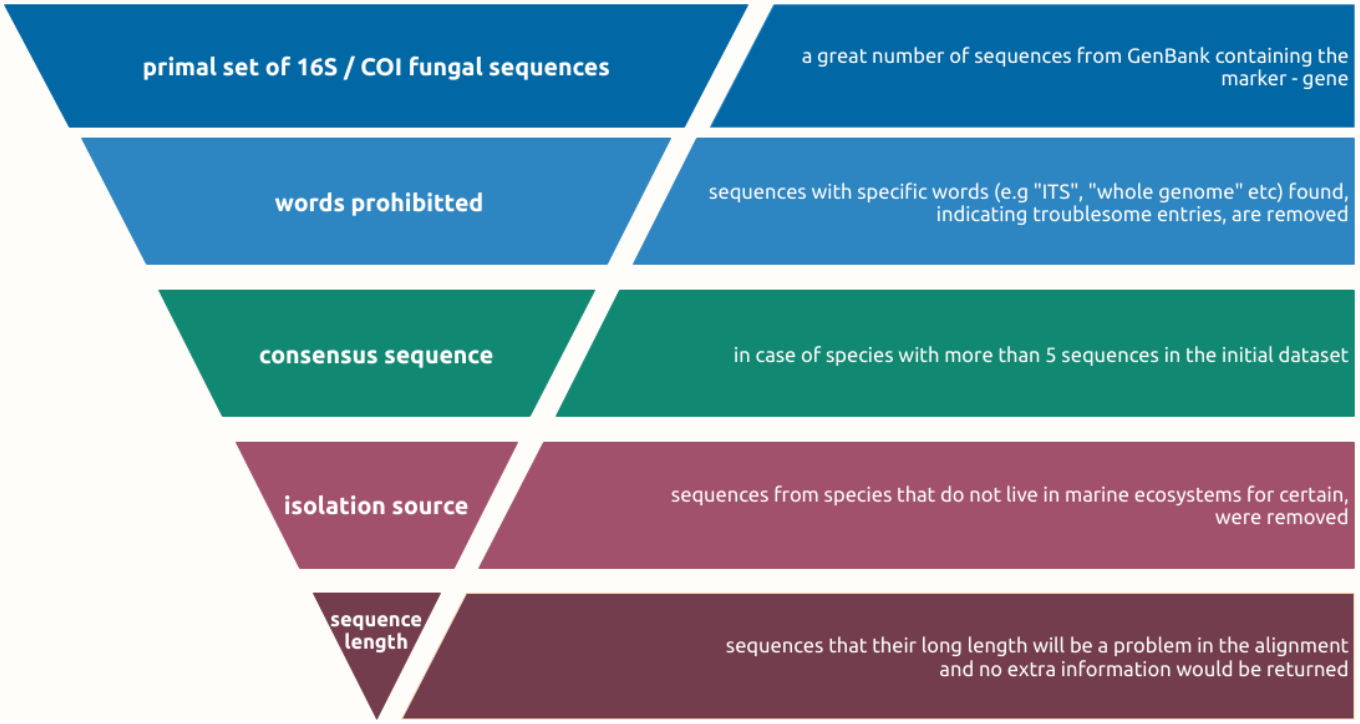


Figure 3: The filtering steps that were followed for the primal sets of sequences.

Furthermore, for the case of 16S and using a Python script, the species for which there were more than 5 sequences were found. The number 5 was set arbitrarily; for those species, a consensus sequence was created using USEARCH [40]. USEARCH is a sequence analysis tool that offers search and clustering algorithms that are often orders of magnitude faster than BLAST.

Then, a python script was written in order to get the results that "efetch" (one of the E - utilities programs that downloads records or reports) returns. In the .gb file that was created, a series of information for each of the sequences that remained after the first filtering steps were mentioned. The value of "isolation source" field from each of them was examined and those that were certainly not marine were removed.

A final filtering step, only in the case of COI, was the examination of the sequence length. We noticed that even after the first two steps, there were still a lot of sequences with a length which was forbidden for our experiment. These sequences included not only our target region but others too, without referring it on their title. Hence, using bash commands they were also removed.

2.1.3 Sequence alignment

Provided that the sequences have been filtered, those that are kept are the "pool" in which motifs that can behave as blocking primers can be found. Thus, a sequence alignment is needed in order to search for such possible pairs. For this task, MAFFT [41] was used for both 16S and COI. MAFFT is a multiple sequence alignment program for UNIX-like

operating systems. In case of 16S, MAFFT was run using its default parameters, while in COI the L-INS-i parameter (accurate; for alignment of <∼200 sequences) was preferred.

2.1.4 Finding possible blocking primers

With the assistance of PrimerDesign–M [42], 5 sets of primers that might work as blocking primers were designed. PrimerDesign–M is a tool whose goal is to find primers for PCR, sequencing (including NGS), and other uses. Primers are based on a multiple alignment and optimized to user-defined criteria. These criteria/parameters (Table 3) play a serious role and some of them were tuned according to the needs of our experiment, while others were set as default.

Table 1: The parameters as were tuned in PrimerDesign-M for the case of 16S

Parameter	**Explanation**	**Value (for 16S)**
Region of interest	the region you need to amplify from your alignment	500-1500
Multiple fragments	define if you want to make one pair of primers or a set of primer pairs spanning a longer region of interest	Single
Adaptor to use	whether DNA handle exists and if yes the way that will be attached to the construct	No adaptor
Read length	if you choose "No adaptor", you need to provide a read length which depends on the read length of the system you are designing primers for	2000

A series of other parameters, such as complexity limit, minimum and maximum primer length etc., were set as by default. A parameter worth-mentioning is the “tag option” which allows to choose how to optimize the tags / barcodes. In each case, one of the 3 optimization parameters is optimized, while the other 2 are specified by user input. The user can optimize the bio-barcode tags based on: 1) a desired number of unique tags, 2) a certain length of the tags, or 3) a minimum edit distance. A higher edit distance makes downstream bioinformatic sorting more robust, translating into fewer lost sequences due to ambiguous reads, and also fewer misclassifications of barcodes [43]. However, in our case the default “no tag” option was chosen.

In order to verify whether the motifs found can actually function as primers and do not have issues that are common in oligonucleotides (e.g. self – dimers, hairpins etc.) OligoAnalyzer 3.1 [44] was used. GC content, melting temperature (Tm) as well as the secondary sequences and the delta - G scores were computed. Delta – G is calculated by taking into account the longest stretch of complementary bases, with the Maximum

Delta G value referring to the free energy of the oligo sequence binding to its perfect complement.

2.1.5 Relative position compared to amplification primers

The sequences that were used for the first alignment (i.e. only the downloaded gene sequences), the sequences of the amplification primers as well as the sequences of the predicted blocking primers, were merged onto a .fasta file. The reverse complement of the reverse primers both from the case of amplification sets and from the sets that PrimerDesign – M returned, were found and set in this new .fasta file.

A second alignment was performed with the sequences of the new file and in order to visualize its output AliView [45] was used. AliView is yet an alignment viewer and editor, probably one of the fastest and most intuitive to use, which allowed us to visualize the relative positions among the two pairs of forward and reverse primers.

2.1.6 Evaluation of the predicted primer sets

To evaluate our findings, beyond the primers' relative positions that can be thought as a first quality control, both ratios of false positives (non fungal sequences to which the blocking primers bind) and true negatives (fungal sequences that blocking primers fail to bind to) were estimated. In this step, different approaches were followed for 16S and COI, due to the lack of some important tools and databases regarding the latter gene.

More specifically, in the case of 16S, TestPrime [46] was used to compute coverages for each taxonomic group in all of the taxonomies offered by SILVA [47]. SILVA is a comprehensive online resource for ribosomal RNA (rRNA) sequence data. It provides comprehensive, quality checked and regularly updated datasets of aligned small (16S/18S, SSU) and large subunit (23S/28S, LSU) rRNA sequences for all three domains of life (Bacteria, Archaea and Eukarya). TestPrime relies on the "ARB PT server" to run the *in silico* PCR and all it needs as an input is a pair of primers and the number of permitted mismatches, which it was set equal to 1.

In the case of COI though, an analogous tool is not existing. Hence, a database with all COI sequences registered in GenBank was created using, as mentioned in Section 2.1.1., e - utilities in order to acquire them, and afterwards the "ARB project" was executed. In this set of sequences, the fungal sequences are not included. As a result, a database with over 2.4 million sequences was built. ARB project was installed in HCMR's cluster and ran through X11 Window System - which provides the basic framework for a GUI environment.

Probe design and probe matching require a lot of database searching. Hence, the basic ARB SEARCH ALGORITHM when any (calling) program requires a database scan , it does not execute the job itself but it calls a special "SEARCH_PATTERNS_N_A_BIG_DATABASE_PROGRAM".

This program is called PT_SERVER ('Prefix tree server') and whilst different databases have different PT_SERVERs, two new PT_SERVERs were built, one with the fungal sequences which were used for finding the potential blocking primers and another for the database containing all the COI entries of GenBank. Using the predicted pairs of blocking primers in both new PT_SERVERs and setting each time the number of mismatches allowed between the primer and the sequences that it might bind to, both false positives and true negative ratios were calculated. However, in order to conclude that a pair of primers actually bind to a specific sequence, then both primers have to bind to this sequence. Hence, the sequences to which the forward primer binds were listed, then the same was done for the reverse complement of the reverse primer and finally only the sequences that existed in both lists were kept as a positive "hit".

Table 2 below presents the tools that were used for the designing of blocking primers. Each of these tools was used for one of the steps that were followed in the designing pipeline which is described in the Methods section. Each of them also, it is not unique in its field as there is a plenty of alignment programs, or viewers etc. However, most of the selected tools had some benefits related to others for our case, as it is also referred in Section Methods.

Table 2: The main tools that were used in blocking primers' design

Tool	Usage	Url
Silva	a comprehensive on - line resource for quality checked and aligned ribosomal RNA sequence data	https://www.arb-silva.de
Silva-TestPrime	evaluate the performance of primer pairs by running an *in silico* PCR on the SILVA databases	https://www.arb-silva.de/search/testprime/
Usearch	a unique sequence analysis tool which offers search and clustering algorithms	https://www.drive5.com/usearch/
OligoAnalyzer 3.1	identifies oligonucleotide properties, including melting temperature, hairpins, dimers and mismatches	https://eu.idtdna.com/calc/analyzer
PrimerDesign–M	finds primers for PCR, sequencing (including NGS), and other uses	https:// www.hiv.lanl.gov/content/sequence/ PRIMER_DESIGN/primer_design.html
MAFFT	a multiple sequence alignment program for unix-like operating systems	https://mafft.cbrc.jp/alignment/software/
AliView	alignment viewer and editor	http://www.ormbunkar.se/aliview/
ARB–project	a graphically oriented package comprising various tools for sequence database handling and data analysis	http://www.arb-home.de/
E–utilities (Entrez)	provides access to the NCBI's suite of interconnected databases (publication, sequence, gene, etc.) from a UNIX terminal window.	https://www.ncbi.nlm.nih.gov/books/ NBK179288/

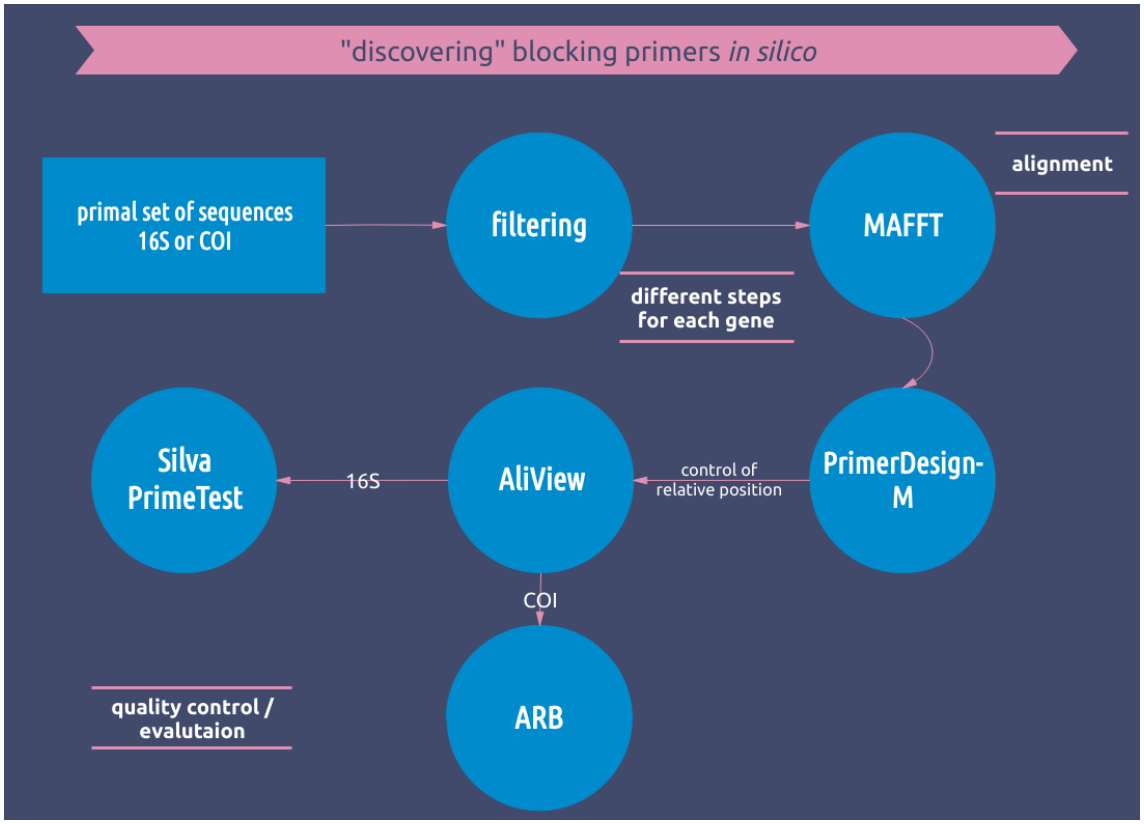


Figure 4: The pipeline that was used for designing blocking primers

3 Results

3.1 *In silico* design of blocking primers

3.1.1 The 16S case

The Entrez query returned 7632 sequences that met the criteria of our query. From each filtering step, a significant number of sequences were removed from our initial dataset. In the table below the removed sequences in each step are shown.

Totally 6998 sequences were removed and the rest 634 were used for alignment. MAFFT's (multiple alignment program) output was an .aln file which was visualized with the aid of AliView.

Table 3: The number of sequences that were removed in each filtering step

Filter	Sequences filtered out	Sequences left
Words - taboo	6629	1003
Consensus sequence	224	779
Isolation source	145	634

Subsequently, the MAFFT's output was used as input for the PrimerDesign - M in order to find oligos within them that could act as primers. The PrimerDesign - M' s results were 5 possible pairs of primers along with their: starting and stopping positions, entropy, complexity and Tm temperature. In addition, PrimerDesign - M produced a pdf file with a graph showing the whole sequence, the region of interest darkened and the positions that the predicted forward and reverse primers bind to the sequence.

In all returned pairs, the forward primer is actually the same sequence. The reverse, on the contrary, is the one that differs but it seems that its binding area is quite specific. Their complexities fluctuate between 4 and 16, with only the 5th pair having complexity equal to 16.

For each pair of the predicted primers a fasta file was created, merging the 16S sequences with both the amplification and the blocking primers (using the reverse complement of the reverse primer). These fasta files were aligned and their results were visualized in order to check the relative position of the amplification and blocking primers.

As a result, we observe that in all cases the forward blocking primer binds before the amplification but close enough and also, the reverse blocking binds after the reverse amplification. We conclude that these pairs can actually be used as blocking primers.

The returned pairs of oligos were then imported in OligoAnalyzer 3.1 in order to examine their potential to be actually used as primers. This way a series of characteristics (hairpin, self-dimer, hetero-dimer) that could not allow such a scenario were checked

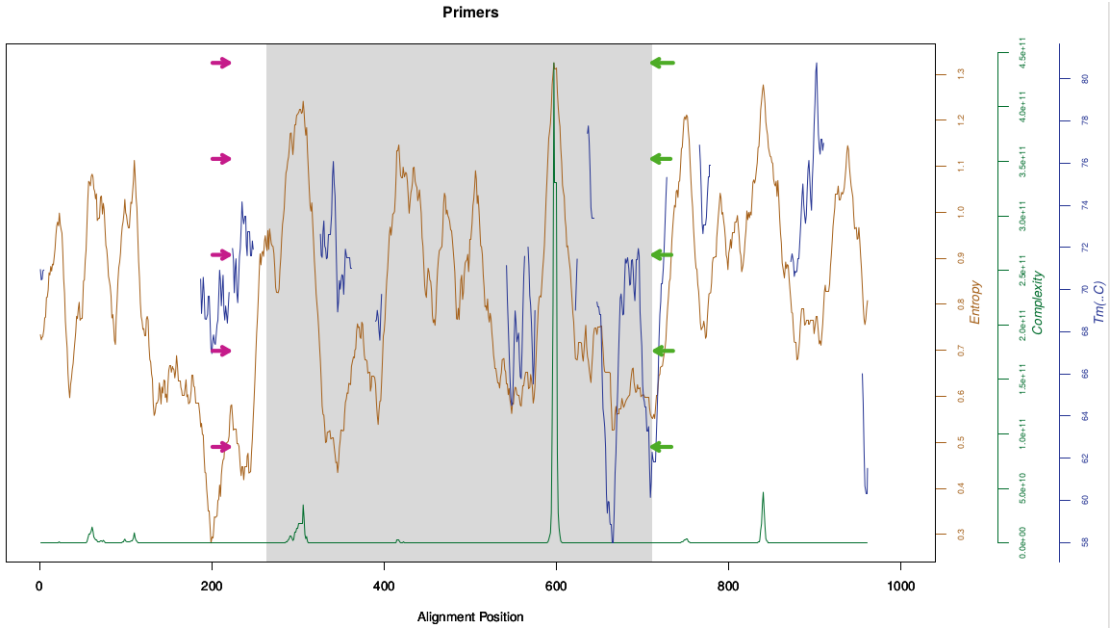


Figure 5: The plot of PrimerDesign - M. The shaded area represents the locus of interest. The red and green arrows show where the designed forward and reverse blocking primers bind to, respectively.

and found not to occur. In figure 5, the results for checking the hetero-dimer of the 5th primer pair (of those returned from PrimerDesign - M) are shown. The value of maximum Delta G equals to -49.62 kclal/mole which was considered high enough.

For all the characteristics for the potential pairs of blocking primers that were checked with the aid of OligoAnalyzer 3.1 the results were acceptable too, hence we could evaluate how efficiently they work as such.

In order to evaluate the predicted oligos as blocking primers, we used Silva TestPrime and the matches of each of the 5 predicted primer pairs with the entries of Silva database were recorded. As shown in Figure 6 based on Silva TestPrime for the 5th primer pair (of those that PrimerDesign - M returned), 82% of the fungal sequences were matched. At the same time, for the same pair of primers we found that there are almost no sequences of Bacteria that match with them, which means that there is almost zero false positive ratio.

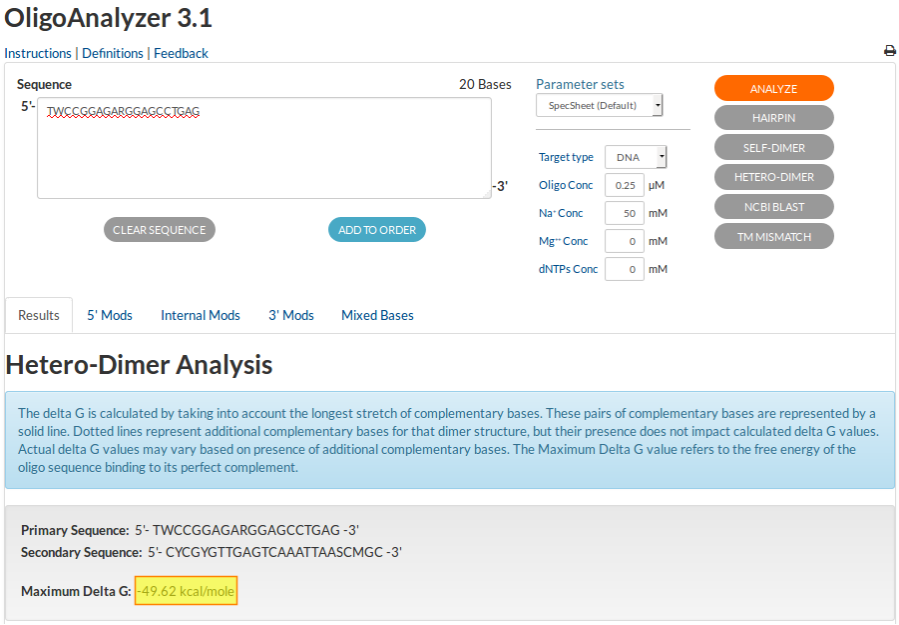


Figure 6: The output of the OligoAnalyzer 3.1 tool showing the Hetero-Dimer Analysis for the 5th pair of potential blocking primers.

The results from the Silva TestPrime for all the other four pairs of primers were also fulfilling, with the best scores deriving from the 5th one.

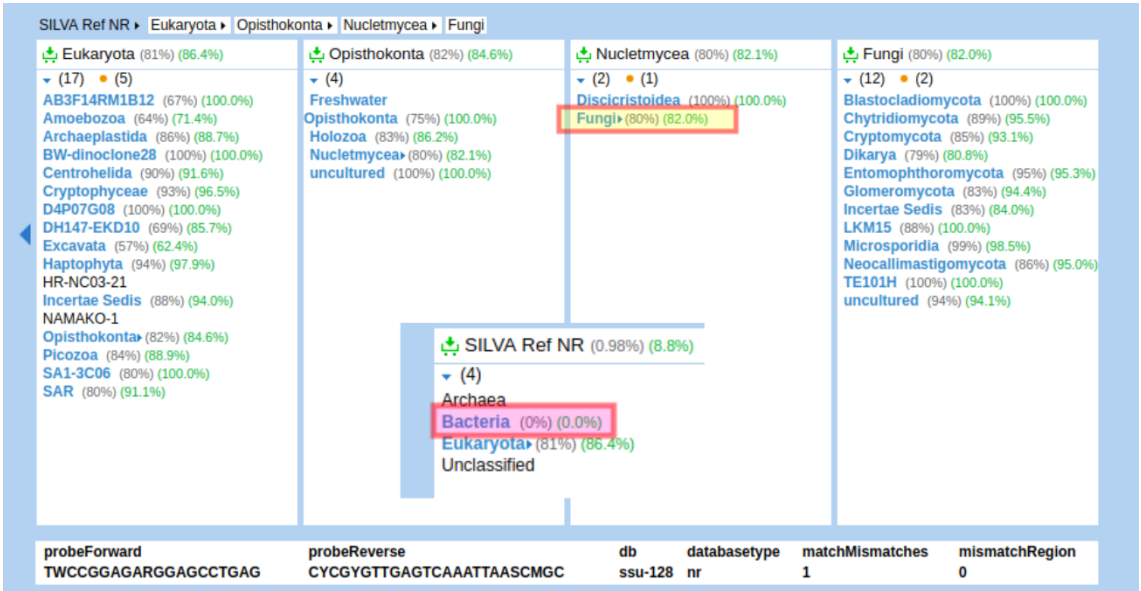


Figure 7: Evaluation of the 16S blocking primers using Silva TestPrime. A significant proportion of the fungal sequences match with the 5th pair of the potential blocking primers (yellow highlight). With purple highlight, the possibility of the primers blocking amplification of bacterial sequences is nearly zero.

As a conclusion, for the 16S gene we ended up with a series of pairs of blocking primers that were worth the effort to be tested in the laboratory, especially the pair:

forward: 5' TWCCGGAGARGGAGCCTGAG 3';
reverse; 5' CYCGYGTTGAGTCAAATTAASCMGC 3'.
In silico designed primers, especially in the case of blocking primers, have to be tested in the laboratory for certainty.

3.1.2 The COI case

In the case of COI gene, the Entrez query returned 2285 sequences. After the filtering steps, the number of sequences were reduced to 652. Compared to the 16S case, in COI sequences an extra step was implemented. Sequences longer than 2500 bp were removed, which resulted in 71 sequences being removed this way. This step was important because these sequences had no information neither on their title nor in their meta - data that could inform us that they do not fulfill our criteria; however, their length would be a serious problem in the alignment step. MAFFT was used for the alignment both with and without these 71 sequences but only the second alignment was used in the next steps of our pipeline for finding blocking primers.

MAFFT's output alignment was used as input in the PrimerDesign - M. In this case, contrary to 16S, it was the reverse primer that had the same position in all the pairs that were returned, as it is shown and in the table below (Table 4). Complexity and Delta G of pair value were also returned by the PrimerDesign - M tool, with the higher complexity being 8.

Table 4: The output of PrimerDesign - M for the case of COI gene

Set	Primer Constructs	Start	Stop	Entropy	Complexity	Tm(C)	Tm SD (C)	Tm Range(C)	Maximum Delta G	Delta G of pair
1	F1.1 5'CCWGATATGGCATTYCCTAG	168	187	0.06731	4	55.9	1.16	54.89-56.90	-36.95 kcal/mole	
1	R1.1 5'CATMCAAATAAAGSTAATTTRTG	457	435	0.08863	8	51.67	1.87	49.05-54.27	-38.33 kcal/mole	-38.33
2	F2.1 5'TTTGGTAATTWYTTATTACC	132	151	0.06859	4	46.26	1.06	44.89-47.36	- 33.08 kcal/mole	
2	R2.1 5'ATMCAAATAAAGSTAATTTRTG	456	435	0.09266	8	49.83	1.96	47.08-52.56	-36.38 kcal/mole	-36.38
3	F3.1 5'ATGATWTTYTTYATGGTTATGCC	93	115	0.07574	8	54.72	1.42	52.56-56.87	-40.66 kcal/mole	
3	R1.1 5'CATMCAAATAAAGSTAATTTRTG	457	435	0.0886	8	51.67	1.87	49.05-54.27	-38.33 kcal/mole	-40.66
4	F4.1 5'GATWTTYTTYATGGTTATGCCTG	95	117	0.0757	8	55.76	1.43	53.58-57.93	-40.78 kcal/mole	
4	R1.1 5'CATMCAAATAAAGSTAATTTRTG	457	435	0.08863	8	51.67	1.87	49.05-54.27	-38.33 kcal/mole	-40.78
5	F2.1 5'TTTGGTAATTWYTTATTACC	132	151	0.06859	4	46.26	1.06	44.89-47.36	-33.08 kcal/mole	
5	R5.1 5'TMCAAATAAAGSTAATTTRTG	455	435	0.0971	8	48.96	2.04	46.09-51.81	-34.9 kcal/mole	-34.9

Subsequently, a second alignment was created, having as input the sequences that were used in order to find the blocking primers, merged with the amplification primers for COI gene and the predicted blocking primers (Table 4). Their relative position was then checked, with both forward and reverse primers being placed as they should be, i.e. the forward blocking placed before amplification forward primer and the reverse blocking primer after the reverse amplification one.. However, the reverse primer seemed not to bind with a number of sequences, which will be discussed further in the evaluation step.

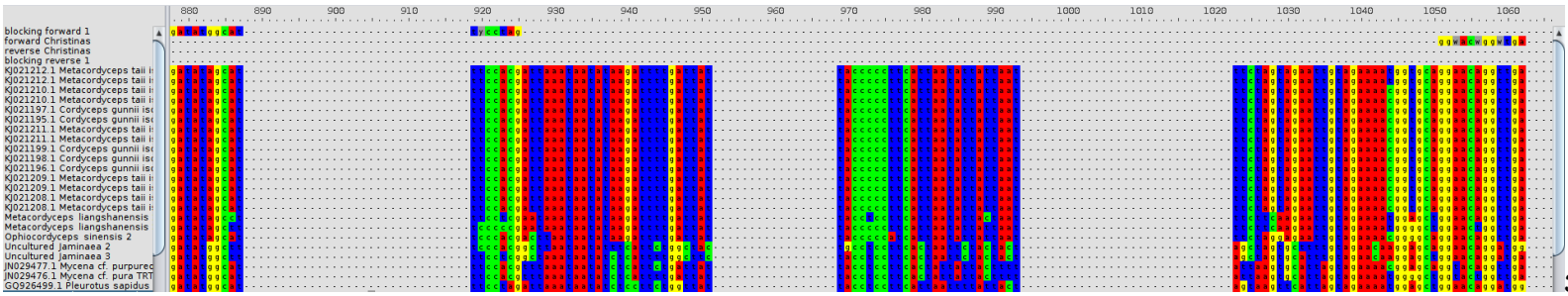


Figure 8: Alignment' s visualization using AliView. The relative position of the two forward (amplification and blocking) primers are shown.

OligoAnalyzer 3.1 was then used again in order to check the characteristics of the oligos returned from the PrimerDesign - M and it was proved that they can be used as primers indeed as for all the characteristics examined (hairpin, self-dimer, hetero-dimer) the returned values were within the permissible limits.

In order to evaluate the potential COI blocking primers, and due to the lack of an already available tool analogous to Silva TestPrime, a database with all COI entries in GenBank was created and a new PT_SERVER in ARB as well (non - fungal COI database). Because of the enormous number of sequences (over 2 millions), this database was built in two parts. In addition, another database and another PT_SERVER was made with the fungal sequences that were used in order to design the blocking primers (fungal COI database). We then used the PROBE MATCH tool in order to find the matches between the potential primers and the sequences of the two databases. PROBE MATCH cannot match degenerated oligos hence we produced all the possible combinations and we summed up the matches that derived from a single degenerated primer. The matches

from all the different oligos that can be produced from each degenerate primer were matched and then the names of the matched sequences for each of them were kept. Hence, for each primer pair we ended up with two files with sequences' names, one for the forward and one for the reverse. In order to find with how many of the sequences of each database the primers bind to, the sequences that existed in both these files were counted (see Appendix).

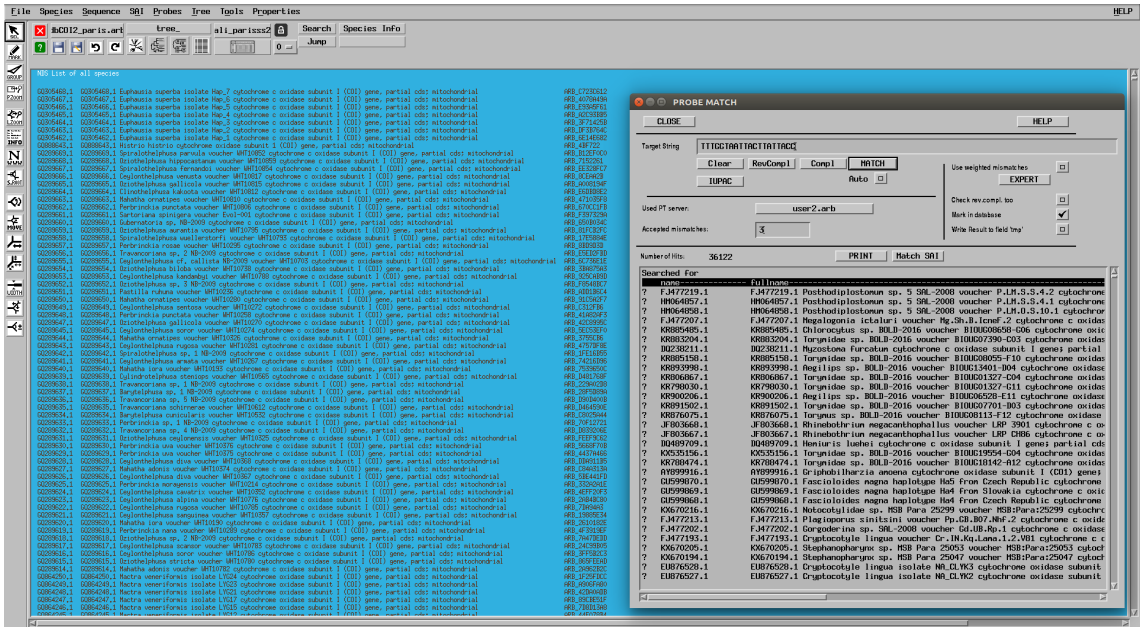


Figure 9: ARB database and PROBE MATCH tool in ARB. One of the primers that can be produced from the degenerate forward blocking primer (of the 2nd pair that was returned from PrimerDesign - M) is matched with the second part of the non-fungal COI sequence database.

Finally, the binding proportions between the designed primers and the fungal COI sequences, as well as the non - fungal, were calculated and plotted using Rstudio. As shown in Figure 10, all 5 pairs have a fungal binding percentage approximately equal to 70%. The false positive ratio is rather small for all of them. However, the 4thpair of blocking primers (forward: 5' GATWTTYTTYATGGTTATGCCTG 3'; reverse: 5' CATMCAAATAAAGSTAATTTRTG 3') could be tested in the laboratory, as it seems to have the best behaviour with 72% successful binding to fungal COI and less than 0.5% false positive ratio, i.e. binding to non-fungal COI sequences.

Similarly with case of 16S gene, a series of*in silico* designed primer pairs should be tested in the laboratory in order to have certain results on whether they can be used as blocking primers, allowing the amplification of every other organism except Fungi, or not. In every case, the evaluation of our returned primers for both genes (16S and COI) allows us to be optimistic, but final confirmation of the *in silico* predictions will be provided from the laboratory experiments.

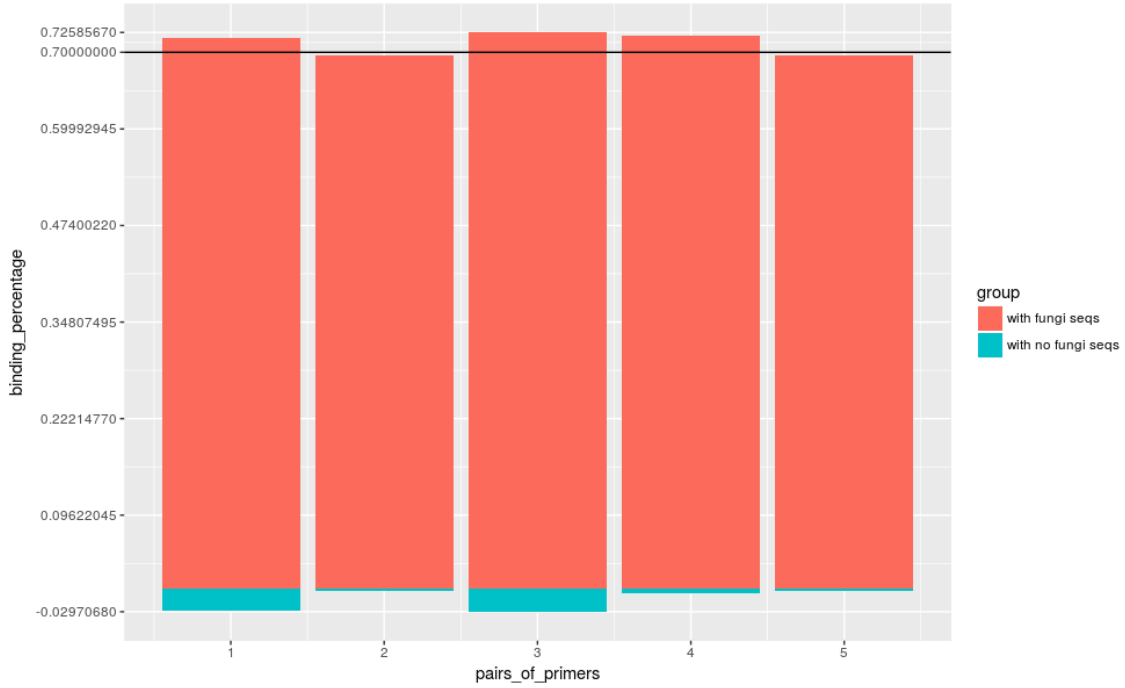


Figure 10: Evaluation of the COI blocking primers using the custom made database for the needs of this project. The graph is showing the binding percentages to the fungal (red) and to the non-fungal sequences, i.e. the false positive ratio (blue).

4 Discussion

4.1 Is it commonplace to design blocking primers for groups?

It would not be an overstatement to say that the undertaken task was not that elementary. A small motif that exists in the fungal genes and not in any other organism had to be found and simultaneously this motif had to be located very close to the chosen amplification primers; as a matter of fact, ideally, the forward blocking primer should be before the forward amplification primer and the reverse blocking after the reverse amplification. Such conditions are quite difficult to occur simultaneously, which is why in most cases blocking primers are species specific and target DNA of one and only species in order to prevent its amplification.

For example, gut contents of semi-digested prey homogenate contain highly degraded prey DNA mixed with abundant high-quality DNA of the predator itself. Therefore, predator DNA co-amplification may prevent or bias prey recovery if no preventive measure is taken. Consequently, predator-specific annealing blocking primers should be used (tailed mlCOIintF and jgHCO2198) [48].

In addition, two genes were selected to be examined. This is because in metabarcoding protocols, due to the lack of one global barcode for all organisms, there are different

marker genes depending on the target group that needs to be analyzed. Hence, our attempt was to design a blocking primer for 16S in order to prevent the amplification of Fungi when microbes were those under question and for COI when Eukaryotes were being investigated.

4.2 Are our findings worthy of trust?

Regarding 16S gene the obtained results are more than encouraging. Two significant traits occur of those potential blocking primers that can make us feel positive to their true behaviour as such. First, the fact that they bind to the fungal sequences with the ideal way, i.e. their relative positions with the amplification factor are optional. And second, as Silva TestPrime indicates, they manage to match with a high percentage of Fungi and at the same time to not bind with (almost no) microbes. As a result, there is certainty that no bacterial species will be missed from our analysis due to a false match between the blocking primer and a non - target sequence. Also, if there is any fungal sequence in the sample, then it is quite possible that the designed blocking primer will match with it and prevent its amplification.

When it comes to COI gene, the returned results were not that promising. And that is mainly because of the relative position of the reverse blocking primer. As the evaluation step revealed, there are only a few non fungal sequences to which the designed blocking primers binds. Hence, this pair of blocking primers can be checked in the laboratory without the concern of not letting the amplification to occur in organisms of interested. However, there is no certainty that actually the fungal COI amplification will not be blocked.

In order to evaluate in an absolute way the designed blocking primers, they should be tested in the following experiment: metabarcoding analysis from our samples both with and without using them. In each case - 16S and COI - a significant difference between the two amplification conditions should be observed, with a number of fungal species being absent in the results of the first and present of the second's. Only in this way it would be possible to conclude whether or not our findings are valuable as blocking primers.

4.3 The need of a "COI version" of Silva TestPrime

The non - existence of a tool similar to Silva for the case of COI turned out to be a very serious problem in our research. As mentioned in Methods, the SIlva TestPrime works for the 16S with almost exactly the same way as the circumvention that was followed in the COI case, since it also uses the ARB PT_SERVER. However, it was rather time

consuming to retrieve all the COI sequences from GenBank, create a database and then a PT_SERVER. In addition, as ARB needs a GUI, connection through X11(a network protocol for Unix and other operating systems like this, to allow graphical access to applications remotely [49].) was necessary, as well as retaining online status for quite a few hours until the database was created. Lot of technical difficulties had to be dealt with at this step; hence, it is our belief that a "COI version" of Silva is more than needed. As mentioned in their website, the BOLD Identification System (IDS) for COI accepts sequences from the 5' region of the mitochondrial Cytochrome c oxidase subunit I gene and returns a species-level identification when one is possible. That means that BOLD does have a complete and updated database of the COI entries; however, this can be used only to identify a specific COI sequence (with minimum length equal to 80bp) to a specific species. At the moment it cannot be used for primers but an extra choice could possibly be added in the BOLD project in the future.

Part III

P.E.M.A: a Pipeline for Environmental DNA Metabarcoding Analysis

1 Introduction

1.1 Millions of reads: what could be done with so many of them?

Metabarcoding opens a new era in biomonitoring and biodiversity. However, from the Illumina output to an amplicon study analysis results, there is a long way to go. For such a task, numerous of steps have to be performed one after another. Even plenty of tools have been developed for each of these steps, there is still not one to implement all of them. As a result, for amplicon analysis there is not a standardised routine yet. Data processing must reflect the peculiarities of each specific marker gene, the kind of technology that was used for the sequencing, as well as the needs for a specific experiment, such as the requirements for sample multiplexing [34].

A plethora of pertinent bioinformatic tools is available. In many cases different tools may employ rather different algorithms for the same task. The different parameters of each tool in each step and the parameters' combinations among the different tools could cause a combinatorial explosion in the number of parameter sets. Hence, the final output from the same dataset (raw data), can differ significantly, according to the protocol followed.

Well-established pipelines are available to process metabarcoding data (RDP – Cole et al. 2009; MOTHUR – Schloss et al. 2009; QIIME – Caporaso et al. 2010;) for the case of 16S marker gene and Bacteria communities. However, these pipelines cannot be used in a straightforward way for metabarcoding analysis of eukaryotic organisms, as adaptation to other marker genes is required. In addition, the pipelines mentioned above, although established they still have a series of difficulties to address such as OTU (Operational Taxonomic Unit) taxonomy assignment.

A metabarcoding pipeline comprises a great number of steps, in each of which a series of different approaches can be executed to overcome the different issues that may occur. In the next paragraphs of Introduction, all these steps, the difficulties in each of them and the algorithms that have been developed so far to deal with them are described. Furthermore, the reasons that led to the algorithms that were finally chosen to be included in P.E.M.A. as well as a brief overview of the way they work are described

later in Methods section.

1.2 Read pre-processing

A significant difficulty in metabarcoding amplicon analysis, is the errors occurring during the amplification or the sequencing step (see Section.4: "NGS Illumina Mi-Seq: a brief overview" paragraph). Library preparation method and the primers used, are the most significant sources of bias and cause distinct error patterns [22]. Due to these errors, an initial dataset of sequences considerably different from the sequences present in the original sample, is produced.

For Illumina, substitution type miscalls is the dominant source of errors. This is due to a strong correlation of A and C as well as G and T intensities. These correlations come from the very much alike emission spectra of the fluorophores as well as the filters' limitations used to distinguish these four signals [22].

Hence, the quality control of Illumina sequencer output is necessary as well as the attempt to fix read errors as far as possible and thus improve their quality.

1.3 OTU clustering

NGS revolutionized Bacteria's at first and then all species's surveillance and therefore, monitoring biodiversity by enabling low-cost, high-throughput sequencing of the 16S gene. OTU clustering methods were applied to NGS output, but it soon was for sure that a great number of artifacts OTUs were generated by these methods due to experimental error. This way, inflated estimates of diversity were concluded [50].

OTU clustering provides a large computational benefit. Millions of reads can be produced during a typical 16S amplicon analysis. Using OTU clustering, this number can be reduced in only some thousands. That is important, especially from a computational point of view, as in a series of downstream analyses, as multiple sequence alignment (MSA) or phylogeny estimation, both computational sources and time are reduced [51].

The size and number of OTUs created can differ considerably according to the clustering algorithm used. The OTU-based methods can be grouped into hierarchical clustering, heuristic clustering and model-based clustering methods. There are two fundamental flaws, the popular *de novo* amplicon clustering methods suffer from: arbitrary global clustering thresholds, and input-order dependency induced by centroid selection [52]. That is why multiple substitutions that may occur at the same site are not taken into account from these clustering algorithms [51].

For these reasons, P.E.M.A. provides 3 different clustering algorithms and the user is able to balance the pros and cons of each according to the dataset and the study aim. USEARCH is a commonly known algorithm for the OTU clustering in 16S amplicon analysis. According to the bibliography, it has been implemented in COI amplicon analysis as

well. Moreover, Swarm and CROP, even in P.E.M.A. they are options for OTU clustering of COI amplicons, in bibliography they both have been used for 16S as well.

1.4 Taxonomic assignment - building the OTU-table

Taxonomic assignment is the process of naming sequences. Typically, one "representative" sequence is selected from each OTU for assignment. However, nothing prevents from individually assigning each sequence read, for example in case of no clustering.

Taxonomic assignment could occur either at species level, or at a higher taxon level (genus, family etc). To perform taxonomic assignment, two factors are needed:
1) a comprehensive marker gene - species mapping database (reference database) the richness of which directly affects the identification accuracy
2) a sequence matching algorithm.

The latter may involve one of the following approaches:
Alignment-based approaches in which a number of different measures of similarity is used to compare the query with the reference sequences, based only on their alignment
Probabilistic approaches: based on likelihood estimates of OTU placement
Tree-based approaches: here the similarity between query and reference sequences is estimated by analysing the position of each individual OTU relative to the reference sequence on the phylogram as well as its bootstrap support
Phylogeny-based approaches: in which full-length sequences of the gene that includes the barcoding region are used to build a manually curated reference alignment which is then used to create a reference phylogeny. Taxonomic assignment is then performed using the query sequences and the reference tree as a constraint. The assignment is made after testing placement of the reads across all nodes of the tree. To do so, for every combination among the query sequences and the reference tree, the placement likelihood is calculated and the highest scoring placements are kept for evaluation [53].

Although the clustering of sequence reads into OTUs has been a matter of major concern, less attention has been paid to the taxonomic classification of (M)OTUs, upon which, biological inference may be based [54]. For example once an (M)OTU-table via taxonomic assignment has been built, a further analyses of the ecosystem under study can be performed using either presence/absence or relative abundance perspectives, for the different taxa that may appear in a system or not.

1.5 OTU-table analysis

All previous steps, once completed, result in an (M)OTU table listing a sampling site taxa. Such qualitative result could support research further if turned quantitative. Despite the

notable progress in environmental DNA metabarcoding analysis, the ability to quantify species relative abundance remains uncertain and as a result, it limits its application for biomonitoring [55].

Species abundance, a means of quantification, is the number of individuals per species, while relative abundance refers to the evenness of distribution of individuals among species in a community, [56] that means how common or rare a species is, relative to other species in a defined location or community.

Although a hard task a positive relationship between species abundance and sequencing read abundance has been observed, in case of studies that target specifically a number of taxa [57].

As it concerns the analysis of the OTU-table to make conclusions about the ecosystem under study, a series of tests and metrics have been developed for Bacteria communities, that allow an holistic perception of it. Yet, this is not the case when Eukaryotes are under study. Microscopic Eukaryotes (meiofaunal metazoa, Fungi, microbial Eukaryotes, eggs and juvenile stages of some larger metazoan species) are abundant, diverse, and fill critical ecological roles across every ecosystem on earth. However, there is still a notable gap in our understanding of their biodiversity [58] . Despite this, a series of new tools are available and many useful ecological analyses can now be performed such as: community summaries, heatmaps displaying OTU abundance across sample sites, alpha diversity, phylogenetically-informed beta diversity and ordination, and OTU network analysis [58].

1.6 Computational power and time needed

A major difficulty in a metabarcoding analysis is the computational power and data storage space needed. A number of factors can increase both dramatically. The size of the dataset, the tools used in the pre-processing step, the clustering algorithm as well as the algorithm used for the taxonomic assignment, along with the potential of a tool or not, to be performed in a parallel way. For this reason, this kind of analyses are usually performed on servers or even High Performance Computing (HPC) systems (clusters in this thesis). Even then, long computations times are to be expected.

1.7 Aim of P.E.M.A.

P.E.M.A (Pipeline for Environmental DNA Metabarcoding Analysis) attempts to merge state-of-the-art bioinformatic tools for all necessary steps of amplicon analysis. From the Illumina sequencer output to the analysis of eDNA samples to make conclusions for the ecosystem they came from, a series of tasks need to be implemented. Pitfalls are present in each and every step of the way and the further improvement of the algorithms used is certain. Hence, P.E.M.A. is an “open” project and there is a long way for that to be completed.

2 Methods

Purpose of the P.E.M.A pipeline is to make an (M)OTU table out of the raw data that the user gives as input for two marker genes, 16S and COI. Two datasets were selected to test P.E.M.A., one for each of the two marker genes. P.E.M.A.'s parameters were set up according to the needs of each dataset. P.E.M.A. comprises four main parts taking place in tandem, which are presented below.

2.1 P.E.M.A. in a nutshell

In the first step, the fastq files generated by Illumina MiSeq runs and submitted to P.E.M.A. as input, undergo a quality check. Then, they are processed with criteria associated with the primers used in the amplification step, as well as other sequencing parameters. This pre-processing is indispensable for the metabarcoding analysis of the raw data to proceed. In the second step, the processed reads are clustered in OTUs. Three different clustering algorithms are supported by P.E.M.A. In the third step, the OTUs that emerged from the previous step, are assigned to taxonomies and the phylogeny-OTU table is created. An OTU/MOTU (Molecular OTU) might be assigned to a species or to a higher taxon. Finally, in the fourth P.E.M.A's step, 16S results undergo further biodiversity-profile analysis. The storage, organization and management of the analysis final results takes place in this step.

In the figure below (Figure 11), the overall P.E.M.A. architecture is presented; the numbered quadrants correspond to the steps briefly mentioned above, and to the main Method sections that follow.

2.2 Quality control and pre-processing of raw data

2.2.1 Quality control for the high throughput sequencing (HTS) reads - FastQC

FastQC [59] was used for the read quality assessment. FastQC is a well-known program that conducts a series of quality control tests to check the quality of raw sequence data, deriving from any type of HTS. It determines whether data are problematic or further analysis can be performed. To this end, FastQC runs a set of analyses on each and every raw sequence file (fastq) and produces a report which summarizes the results. FastQC is the first tool invoked by P.E.M.A. This way, after the completion of P.E.M.A.'s first step for a certain dataset, the user has a complete overview of the data. This way, samples whose .fastq files are of low quality can be excluded from subsequent P.E.M.A. runs of this particular dataset. However, removing files completely does not secure that all the remaining reads are of satisfactory quality, as either smaller regions of a read or specific-position errors might have occured during the sequencing. The next tools in P.E.M.A. address these issues.

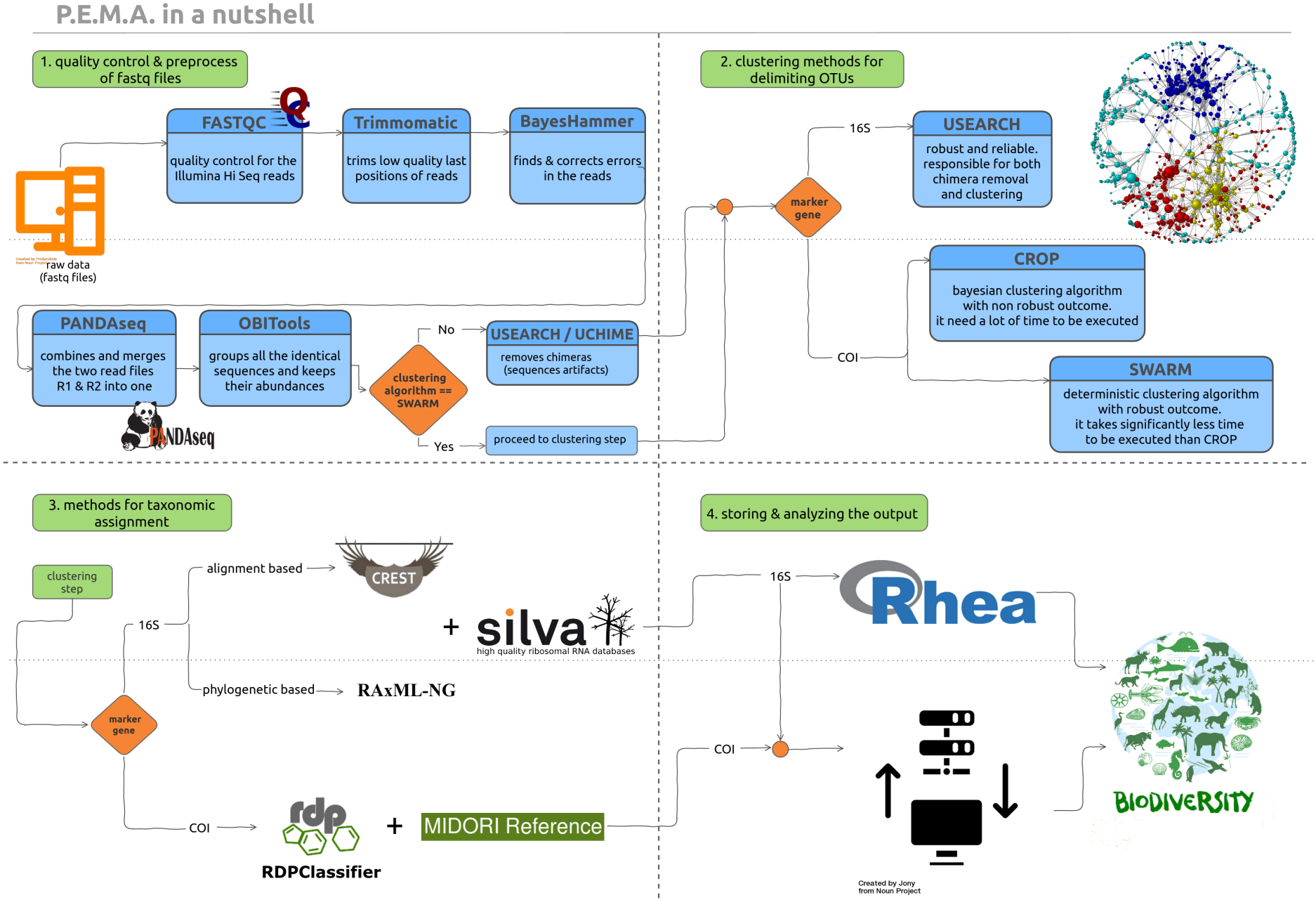


Figure 11: P.E.M.A in a nutshell

2.2.2 Trimming low quality positions of reads - Trimmomatic

It is known the quality of reads decreases as the sequencing progresses [60]. In an attempt to mend the errors that are produced by the Illumina sequencer, a series of different tools have been added in P.E.M.A.

Trimmomatic [61] is able to perform a variety of useful trimming tasks for both single- and paired-end reads. These involve two main tasks: 1) the removal of technical sequences that might have been produced due to the adaptors or the primers sequences and 2) quality filtering, where based on the quality score of each base position, it is determined whether the read should be trimmed.

Hence, Trimmomatic includes a series of different ways to truncate the initial reads. Depending on the parameters set in the sequencing procedure, the parameters of Trimmomatic have to be defined accordingly.

As P.E.M.A. deals with amplicons, the handling of technical sequences erroneously produced during the sequencing is crucial. Amplicons in particular could cause significant alterations due to their common presence in the ends of the reads. Trimmomatic addresses the need of removing such sequences.

When run in 'Simple mode', Trimmomatic aims to find approximate matches between the reads and the technical sequences supplied by the user. This way, technical sequences can be identified in any location or orientation within the reads. A considerable overlap between the read and a technical sequence is required to prevent false-positive findings. However, in some cases, the sequenced DNA fragment is shorter than the read length. For these cases, Trimmomatic's 'Palindrome mode' uses both reads and, given the latter are reverse complements, is able to locate short partial adapter sequences; sequences that would otherwise remain undetected as they miss the minimum required overlap.

Trimmomatic, in the above two ways, detects and removes regions of low sequencing quality. However, it is not able to correct a wrong read base. For this task the next algorithm was recruited.

2.2.3 Seeking and fixing specific-position errors – BAYESHAMMER algorithm

Except of the regions of sequencing with low quality, there is also a significant number of specific-position errors in the Illumina MiSeq output, where a particular base in a certain position, has not been determined correctly.

BAYESHAMMER algorithm, an algorithm of the SPAdes assembly toolkit [62], is capable of finding exactly this kind of errors and attempts fixing them. BAYESHAMMER tries 1) to find read k-mers whose sequencing quality is above a predefined threshold ("solid k-mers"), and 2) select the sequencing reads that can be rebuilt based on the

aforementioned “solid k-mers”. Thus, after finding the “solid k-mers”, BAYESHAMMER can decide whether a base is sequenced properly or not [63].

2.2.4 Merging the two read files into one – PANDAseq

After the base correction, it is essential to merge the forward and reverse reads into one. This is far from a trivial task. Naive assembly would be a first approach. However, as a naive approach would search for perfect matches in regions of overlap, many sequences would be discarded because of uncalled or miscalled bases [64].

On the contrary, PANDAseq [64] tries to achieve the best possible assembly of the overlapping paired-end reads. To do that, PANDAseq attempts to correct possible errors using a probabilistic approach based on the overlap data from the paired-end reads. Further, it uses intrinsic properties of Illumina sequencing, including the low probability of gap inclusion. Finally, it also takes into account the primers that have been used.

PANDAseq supports more than one merging algorithms. ‘Pear’ and ‘simple_bayesian’ are the ones used most commonly. In P.E.M.A. the default merging algorithm is set in ‘simple_bayesian’; shifting to another PANDAseq-supported algorithm is possible for users that would like to optimize this step via experimentation.

2.2.5 Dereplication: grouping identical sequences & keeping their abundances – OBITools

In a sample-specific set, some sequences might occur multiple times. Such sequence multiplicity might be attributed to a number of reasons. First, it could be a result of the PCR-based amplification. Second, for species specific sequences, it could reflect the abundance of the species in a sample. Third, in the case of multicellular organisms, it could describe the gene copies. No matter the case, all three aforementioned cases are important for downstream environmental community analysis Hence, it is not only needed to group all the identical sequences in every sample, but also to keep a track of their abundances. For that purpose, P.E.M.A. uses the aid of OBITools [65] and specifically of ‘obiuniq’ program.

OBITools is a set of python programs developed to simplify the manipulation of sequence files. especially in the context of NGS-based DNA metabarcoding [65].

P.E.M.A. uses the ‘obiuniq’ program of OBITools with the PANDAseq merged files. A set of fastq files is returned. In the header of each entry, except from the sample name and the sequence id there is also the number of copies of this sequence that were found in this sample.

2.2.6 Chimera removal: excluding sequences artifacts – USEARCH or VSEARCH

The sequences produced by the aforementioned steps can be considered of high quality and dereplicated. However, they might still contain sequencing procedure artefacts, such as chimeric sequences. The latter need to be handled before further analysis.

P.E.M.A. provides users with the option of selecting among a number of (M)OTU sequencing algorithms. Depending on the clustering algorithm that will be used, the chimera removal step can be executed by P.E.M.A. either before, after or even during clustering. Depending on the specifics of each algorithm, some clustering methods are affected by chimeras while others are not. In the case of COI, the algorithm 'uchime3_denovo' of the VSEARCH package [66] is executed and depending on the clustering algorithm that the user has chosen, chimera removal is performed before (CROP) or after (Swarm) clustering step. However, in case that 16S is the marker gene understudy , the chimera removal is conducted by the same algorithm that implements the clustering - USEARCH [40].

This step completes the raw data pre-processing and P.E.M.A. may proceed to the next step, the clustering of the reads to OTUs. First of all, it is necessary that reads from all samples get merged into one file, maintaining the sample id of each.

2.3 Clustering methods for delimiting OTUs

After the quality control and the pre-processing of the reads, the clustering step is next. A series of clustering methods to group the reads into OTUs, are provided by P.E.M.A. The exact method to be used depends on the marker gene and the user dataset.

2.3.1 16S case - USEARCH / UPARSE-OTU algorithm

USEARCH [40] is a sequence analysis tool that includes search and clustering algorithms that are often orders of magnitude faster than BLAST. UPARSE-OTU algorithm [67] is one of the algorithms that USEARCH contains and constructs a set of OTU representative sequences from NGS amplicon reads.

UPARSE-OTU uses a greedy algorithm to identify a set of OTUs as representative sequences. As high-abundance reads are more likely to be correct amplicon sequences, they are more likely to be true biological sequences. For that, UPARSE-OTU algorithm sorts reads in order of decreasing abundance. This means that OTU centroids tend to be selected from the more abundant reads.

P.E.M.A. uses also the 'sortbysize' algorithm of USEARCH to sort the reads in order of decreasing abundance, and then executes the UPARSE-OTU algorithm. Finally, it executes the 'otutab' algorithm of USEARCH. This way, an OTU table with no taxonomy information attached to it is being built.

USEARCH and UPARSE-OTU algorithms have been adopted by the microbial ecology community and are used in a number of applications concerning the the case of 16S marker gene. However, there is no equivalent for the COI marker gene case. P.E.M.A includes two clustering algorithms for COI, Swarm and CROP. These are presented next.

2.3.2 COI case - Swarm

Swarm is a novel and open source amplicon clustering program that produces fine-scale molecular operational taxonomic units (OTUs), free of arbitrary global clustering thresholds and input-order dependency. Swarm is regarded as a fast algorithm and also its output is robust, a thing that is not that common in *de novo* clustering algorithms [68] [68].

At first, Swarm clusters iteratively, amplicons that are almost identical. A local threshold (d) is used with d being one of the most important parameters of Swarm. Subsequently, Swarm refines its initial results by using clusters' internal structure and amplicon abundances. This approach reduces the influence of clustering parameters and produces robust operational taxonomic units in a fast and way [68].

The way Swarm builds OTUs should not be affected by chimeras, according to the developers of the algorithm. Chimeras will very likely form independent OTUs. Hence, P.E.M.A. checks for chimeras after clustering when Swarm is used. Swarm creates a fasta file containing OTU representatives and with this file P.E.M.A. runs a chimera check, using again USEARCH algorithm [69]. With the output of 'uchime3_denovo', P.E.M.A. builds an OTU-table without taxonomies.

Swarm algorithm was performed in case of Bista et al. (2017) dataset using 3 different values of the parameter d. When d=1, then the parameter 'fastidious' was also selected. That is an extra option that Swarm v2 provides and it is for a second clustering pass to reduce the number of small OTUs. In fact, it creates virtual amplicons, allowing to graft small OTUs upon bigger ones. For Swarm, "small OTU" is one with mass of 2 or less. The different values of the d parameter were used in order to check which option would produce results more similar/comparable to the ones of the original publication, since the authors had chosen USEARCH as a clustering algorithm.

2.3.3 COI case - CROP

CROP - Clustering 16S rRNA for MOTU prediction, is an unsupervised nucleic acid sequence clustering algorithm. CROP perceives the OTUs as a Gaussian mixture and models the clustering process using Birth-death Markov chain Monte Carlo (MCMC) [70], making the MOTU prediction more accurate. CROP adopts an unsupervised probabilistic Bayesian clustering algorithm and uses a soft threshold for defining the OTUs, which is probably more accurate for reflecting real species diversity. It also reduces the effects of PCR and sequencing errors in inferring OTUs (Wei et al., 2016), by clustering the se-

quences affected by these errors to their mother sequences, instead of removing them altogether.

CROP suffers from requiring time-consuming calculations, even when using High Performance Computing (HPC) systems. Moreover, given the heuristic nature of Bayesian algorithms, the reproducibility of CROP can be low, so that different runs on the same input dataset might result in variable numbers of resulting molecular operational taxonomic units (MOTUs) (Wangensteen and Turon, 2015).

2.4 Methods for taxonomic assignment

As presented above, P.E.M.A. executes one of previously presented clustering algorithms, depending on the marker gene and the user dataset. After calculating the OTUs that exist in an experiment, the crucial taxonomy-assignment step follows. To the best user and marker gene analysis support, P.E.M.A. includes three different approaches for this step. An ()OTU table presenting both the taxonomies and the relative abundances thereof is the outcome of this step.

2.4.1 Alignment-based assignment in 16S case - CREST SILVA

CREST is a set of resources and tools for generating and utilizing custom taxonomies and reference datasets for classification of environmental sequences [71]. CREST uses an alignment-based classification method - 'LCAClassifier' algorithm - with the lowest common ancestor algorithm. It also uses explicit rank similarity criteria to reduce false positives and identify novel taxa [71] [72]. The 'LCAClassifier' algorithm is used for the classification of sequences aligned to the reference databases provided. CREST can handle 4 databases: SILVA [73], Greengenes [74], Unite [75] and amoA [76]. Moreover, it allows the user to build his/her own database.

P.E.M.A. uses CREST along with the SILVA database as it is recruited in the taxonomy assignment of OTUs that came out of reads of the 16S marker gene.

2.4.2 Phylogeny-based assignment in 16S case - RAxML-ng & SILVA

To get a phylogeny-based identification of query sequences, first it is necessary to build a reference tree from a manually curated reference alignment using full-length sequences of the gene that includes the barcoding region. That is a key-step for the phylogeny-based assignment.

As 16S is the marker gene, SILVA_132_SSURef_Nr99 database [73] was used. All non-Bacteria entries of Silva's alignment (Archaea and Eukaryotes) were removed. Then, using 'gappa' [77] (a collection of tools for phylogenetic data handling) and specifically the 'art' algorithm from 'gappa', 10.0000 consensus sequences were generated from the curated reference alignment. Finally, with these 10.000 sequences, P.E.M.A.'s reference tree was constructed using RAxML-ng [78] (figure 12, top)

Taxonomic assignment of the query sequences is possible by using the *ad-hoc* built reference tree and probing placement of query reads across all nodes in the reference topology, with the placement likelihood calculated for every combination. The highest scoring placements are retained for evaluation. For this task, P.E.M.A. uses Evolutionary Placement Algorithm - next generation (EPA-ng, a GTR+GAMMA ML model-based tool, figure 12, bottom) [79].

Prior to EPA-ng P.E.M.A. invokes PaPaRa [80] a program for aligning short reads to reference phylogenies. In P.E.M.A. PaPaRa is employed to combine the OTU clustering output with the P.E.M.A. reference tree and produce a phylogeny-aware alignment required for the EPA-ng execution (figure 12, middle part).

Finally, returns a '.jplace' file. To visualize EPA-ng's output file ('.jplace'), a viewer like the interactive Tree of Life (iTOL) is needed [81]. The taxa in which the OTUs were assigned to are highlighted in a way that reflects each OTU 's abundance.

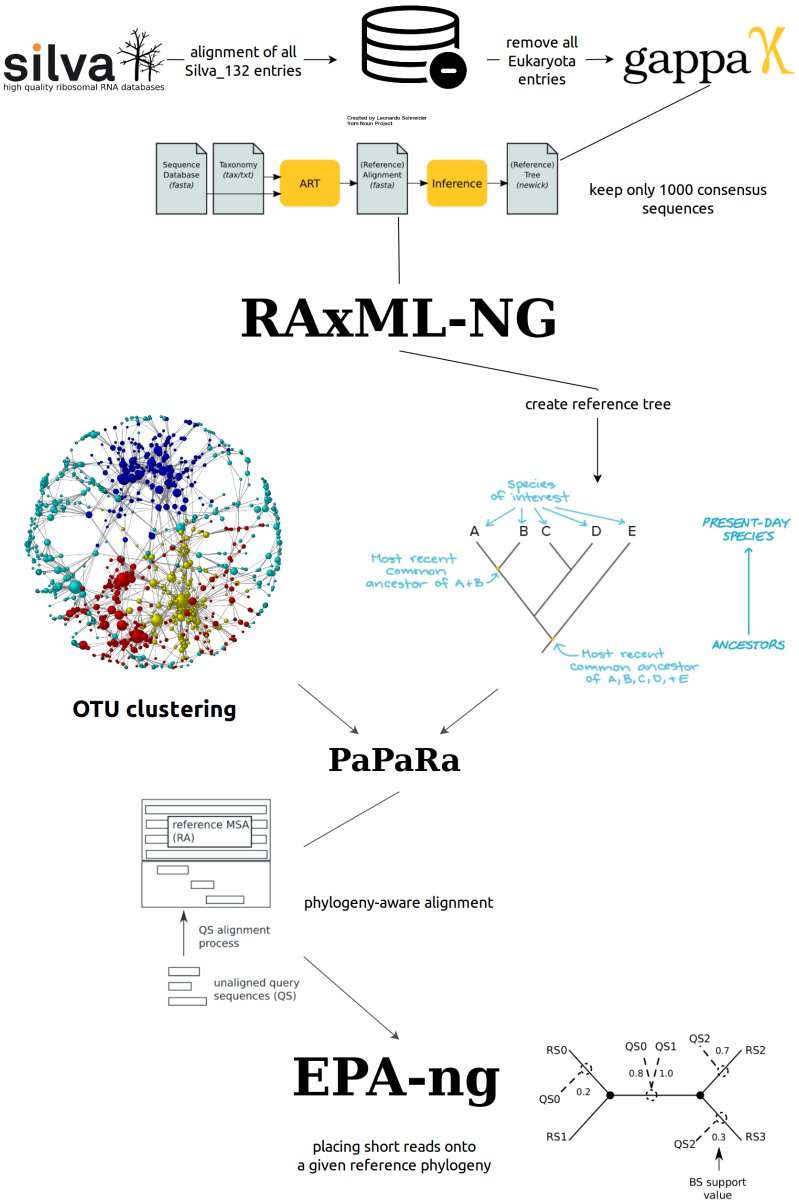


Figure 12: Building the reference tree for the phylogeny-based taxonomy assignment.

2.4.3 Alignment-based assignment in COI case - RDPClassifier & MIDORI database

In case of the COI marker gene, P.E.M.A. includes alignment-based taxonomy assignment only. To this end P.E.M.A. uses 'RDPClassifier', a naive Bayesian classifier originally developed to provide rapid taxonomic placement based on rRNA sequence data. RDPClassifier is a rapid and accurate classifier and it provides taxonomic assignments from domain to genus, with confidence estimates for each assignment [82].

RDPClassifier needs a reference database for the OTU assignment. The MIDORI reference dataset [83] contains all metazoan mitochondrial gene sequences from GenBank, (after quality filtering and taxonomic assignment-tool-compliant re-formatting). P.E.M.A. employs the MIDORI-LONGEST version which contains a single sequence, the longest,

for each species.

2.5 Storing and analyzing the OTU-table

After taxonomy assignment, the final OTU-table is created. In the case of the 16S marker gene, P.E.M.A. provides users with an additional step, the extra OTU-table analysis with the aid of Rhea set of R scripts [84]. Rhea supports the downstream analysis of an OTU table including steps like normalization, alpha- and beta-diversity analysis, taxonomic composition, statistical comparisons and calculation of correlations.

To use Rhea, a tree of the P.E.M.A. derived OTUs has to be built. To this end, the sequences of the final OTUs are aligned using MAFFT so that RAxML can then be used for the tree creation.

Unlike 16S, for the COI marker gene analysis, Rhea-equivalent software has yet to be implemented.

2.6 A user-friendly programming language for developing pipelines - BigDataScript (BDS)

The analyses of large biological datasets often demand, complex processing pipelines that combine numerous tools and may require long execution times even in High Performance Computing (HPC) infrastructures. Simple script-like programming languages developed to orchestrate the bioinformatics tools execution and to manage pipeline data flow are needed. 'BigDataScript' (BDS) is intended as a scripting language for big data pipeline and was chosen for this goal.

BDS, among other advantages, supports the creation of "checkpoints". As a "checkpoint", BDS defines pipeline-state describing files that can be produced either after any task of a BDS-program, in case that the developer has asked to, or when a program fails to be completed. From these checkpoints, the program can be re-executed, exactly from the point that checkpoint corresponds (line on the code), and even on any computer. This way, BDS performs total serialization of the running state and environment. For example, after a failure of a task, the program can be re-executed from that specific point and there is no need to re-run it from scratch. This is also possible in case a pipeline has more than one possible arguments in a specific step. Re-executing the pipeline from a specific step, after changing a specific parameter, is possible with BDS checkpoints. This is quite convenient for a pipeline like P.E.M.A. as after the completion of an analysis, if the user wishes for a single parameter of a single step to be changed, P.E.M.A. can be executed only from the point that this parameter appears and not earlier. For example, even if P.E.M.A has been executed with Swarm as clustering algorithm, BDS checkpoint allow the user to rerun P.E.M.A. with CROP as clustering algorithm, only from the clustering step and forward.

2.7 Datasets for evaluating P.E.M.A.

For evaluating P.E.M.A. two datasets were used, one for the 16S marker gene and one for the COI.

In the case of the 16S marger gene, the dataset used to test P.E.M.A. is retrieved from Pavloudi et al. (2017) [85]. In this study, Remane's "species minimum" concept was investigated for the case of Amvrakikos Gulf. According to Remane, taxonomic diversity of macrobenthic organisms is lowest within the horohalinicum (5 to 8 psu). Sediment samples were collected from 6 different stations (3 riverine, 2 lagoonal and 1 marine); there were 3 replicates samples from each station. DNA was extracted, amplicons of the 16S marker gene were sequenced and metabarcoding analysis was performed.

All samples had the same treatment (i.e. amplicon length, primers etc.). All 18 sample-derived sequence reads were analyzed by P.E.M.A. The analysis results where then compared to those of Pavloudi et al. (2017). The comparison was in terms of OTU prediction and assessment of the Remane's concept validity.

Likewise, in case of COI the dataset used to evaluate P.E.M.A., was from the paper of Bista et al. (2017) [86]. In this project, samples from lakes were collected and two amplicons of the COI marker gene were sequenced, the full-length COI barcoding region (658 bp) and a 235 bp fragment on the 5′ region of the COI barcoding region. The second amplicon was amplified and sequenced for a family-specific study of Chironomidae. In addition, for each amplicon, there were 16 eDNA and 16 invertebrate community DNA samples, i.e. 64 samples in total.

In both cases, the respective fastq files were downloaded from ENA-EBI using 'ENA File Downloader version 1.2' [87].

To assess P.E.M.A.'s result quality and performance, all samples were joint in a single dataset. P.E.M.A. was then run with Swarm as the clustering algorithm used. P.E.M.A.'s output was again compared to Bista's et al., mainly for the total number of species found. From Pavloudi et al. project, a "mapping file", necessary for Rhea's scripts was also retrieved.

P.E.M.A.'s output for these two datasets in a series of different parameter combinations, along with the P.E.M.A.'s statistics were recorded and analysed, in an attempt to evaluate our pipeline.

2.8 Running P.E.M.A.

Running P.E.M.A. in an effective way, relies on the correct parameter choices in the "parameters.tsv" file. As this is an important step, auser may find adequate pertinent information and guidance steps in P.E.M.A.'s manual as well as in those of the tools P.E.M.A invokes.

Setting the marker gene (16S or COI), the name of the analysis (needs to be unique so

as not to be overwritten), the choice of clustering algorithm (in the case of COI marker gene) and of taxonomy assignment (in the case of 16S) are parameters of crucial importance for P.E.M.A. to be executed properly.

Below, (Figure 13), an example of the “parameters.tsv” file is shown, as it was set for the Amvrakikos dataset and the 16S marker gene.

```
##############################################################################################
##############################  P.E.M.A. 's PARAMETERS  ######################################
##############################################################################################
#
#
# In this file there are all the parameters that NEED TO BE ASSIGNED every time you need PEMA to run!
# That does not mean that these parameters that we have her,  are the only  parameters of the tools PEMA uses!
# As you already may know, the combinations are infinite!
# Hence, we encourage you the most to study the manual of each tool and make them as good as possible for your SPECIFIC experiment.
# In each and every variable that you see quotes ( '' ) you have to write inside those the option you choose
# The rest, you need to complete them with a number and never add quotes.
# We chose to have a set of parameters for some tools (mainly for Trimmomatic) as 'by default'. That is because in some cases, plenty of time is required
# in order to have a correct set of those. In the link next to each tool, you can find further information about its parameters.
# YOU NEED TO BE REALLY CAREFUL WHEN YOU FILL THIS FILE !!
#  From each variable you have to leave EXACTLY FOUR (4) BLANKS and then fill the parameter as you wish.
#
## just to test that the parameters are assigned right in our main script:
#
sources      my_parameters_work_just_fine!
#
## give in your each uniq experiment a NAME, so a single output file will be created for each of them
outputFile final_COI_d_10
#
#################################################################
######################  blastqc  ################################
#################################################################
#
## no parameters here!
#
#
#################################################################
###################### trimmomatic  ############################        //  http://www.usadellab.org/cms/uploads/supplementary/Trimmomatic/
TrimmomaticManual_V0.32.pdf
#################################################################
```

Figure 13: Parameter-file of P.E.M.A.

3 Results

Every time that P.E.M.A. is performed, a folder named as the user has determined, is created. In case that the user runs for a second time an analysis with the same name, without moving the first one to another folder, then P.E.M.A.’s output will over-right the first analysis. This folder contains all output files that P.E.M.A. produces and has 8 folders numbered by serial number and one extra, in case that Rhea’s set of scripts, has been performed.

3.1 Quality control and pre-processing of raw data

3.1.1 Step 1: Quality control - FASTQC

As FASTQC is the first tool that P.E.M.A. invokes, the first folder in the output folder contains the quality control results. In the folder with the quality control results, there is a folder for each sample, as well as a .html file and a .zip file which contain all the information included in the folder with the sample’s output.

In the figures below (Figures 14-15) a typical example of FASTQC’s output is shown, from one of the samples of 16S case study. In the summary (Figure 14), the outcome of each of the tests that FASTQC performs is presented. The sequences of each sample, could get either a “pass”, “warn” or “fail” to each test.

```
haris@zorba:~/metabar_pipeline/results/COI_d_1/1.quality_control/ERR1308201_1_fastqc$ more summary.txt
PASS    Basic Statistics        ERR1308201_1.fastq.gz
FAIL    Per base sequence quality       ERR1308201_1.fastq.gz
FAIL    Per tile sequence quality       ERR1308201_1.fastq.gz
PASS    Per sequence quality scores     ERR1308201_1.fastq.gz
FAIL    Per base sequence content       ERR1308201_1.fastq.gz
FAIL    Per sequence GC content ERR1308201_1.fastq.gz
PASS    Per base N content      ERR1308201_1.fastq.gz
WARN    Sequence Length Distribution    ERR1308201_1.fastq.gz
FAIL    Sequence Duplication Levels     ERR1308201_1.fastq.gz
WARN    Overrepresented sequences       ERR1308201_1.fastq.gz
PASS    Adapter Content ERR1308201_1.fastq.gz
```

Figure 14: FASTQC output summary

In addition, a table with some basic statistics of the raw data is shown, as well as a figure with the per base sequence quality of the examined .fastq file (Figure 15). As already mentioned, the quality of the sequencing drops as the sequence length increases. The .html file, contains the output figures of all the 10 FASTQC quality tests.

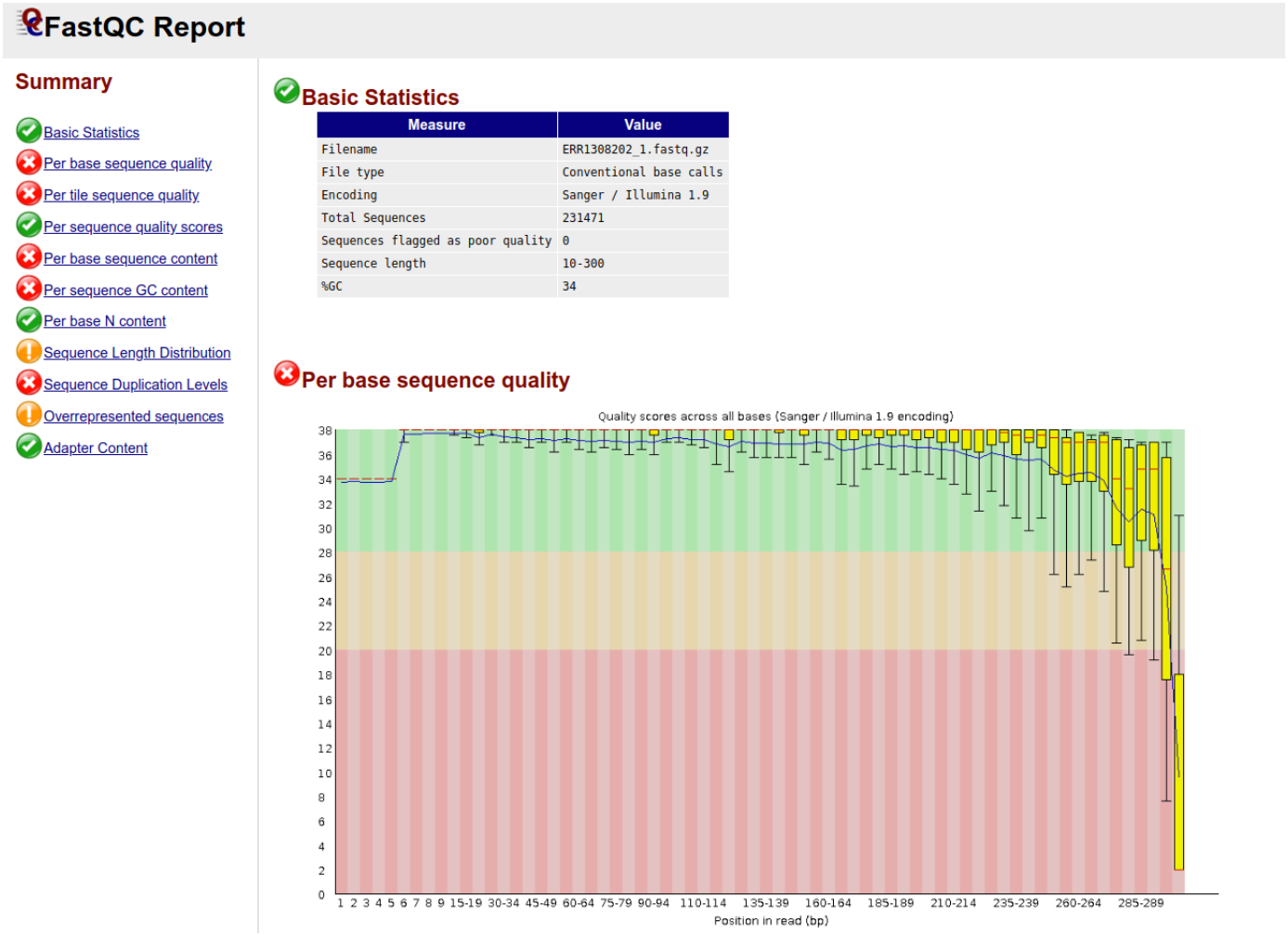


Figure 15: A screenshot of the .html file that FASTQC produces where the results of all tests are shown

3.1.2 Steps 2-6: Pre-processing of the sequences

All pre-processing steps for the sequences of both 16S and COI marker genes are the same. As a matter of fact, the first 5 of P.E.M.A 's output folders, contain the results of the tools invoked in the pre-processing step of P.E.M.A. and are exactly the same in the two marker genes (Figure 16).

```
haris@zorba:~/metabar_pipeline/results/COI_d_1$ ll
total 477539
drwxr-xr-x 130 haris users       386 Αύγ  29 01:41 1.quality_control
drwxr-xr-x   2 haris users       130 Αύγ  28 14:41 2.trimmomatic_output
drwxr-xr-x  66 haris users        66 Αύγ  28 17:36 3.correct_by_BayesHammer
drwxr-xr-x   3 haris users        67 Αύγ  28 18:27 4.merged_by_PANDAseq
drwxr-xr-x   2 haris users        66 Αύγ  28 18:29 5.dereplicate_by_obiuniq
drwxr-xr-x   2 haris users        66 Αύγ  29 00:49 6.linearized_files
drwxr-xr-x   3 haris users         3 Αύγ  28 14:24 7.gene_dependent
drwxr-xr-x  66 haris users        66 Αύγ  29 01:35 8.output_per_sample
-rw-r--r--   1 haris users 581479905 Αύγ  28 18:30 all_samples.fasta
drwxr-xr-x   2 haris users         8 Αύγ  29 01:41 checkpoints_for_COI_d_1
-rw-r--r--   1 haris users 581457163 Αύγ  28 18:30 teliko_all_samples.fasta
```

Figure 16: P.E.M.A.'s output main folder

After each step during pre-processing, the total number of reads decreases. That is because each of these steps (as described in the Methods section) includes a quality threshold. In the table below, the reduction of the reads after the performance of each of the tools invoked in the pre-processing step of P.E.M.A. is shown (Table 5). All sample libraries that start with "L" correspond to samples collected from lagoons, with "S" from the sea while "R" stands for the riverine samples.

Table 5: Number of sequences after each pre-processing step for the case of 16S.

sample libraries	run_accession	initial number of reads	after trimming	after specific-position error correction (Bayes-Hammer/Spades)	after merging (PANDAseq)	after dereplication (obiuniq)
L_LOout_a	ERR1906853	96457	95748	95569	46576	37503
L_LOout_b	ERR1906854	144529	142133	141847	65556	56636
L_LOout_c	ERR1906855	139149	138811	138490	64650	51889
L_LOin_a	ERR1906856	130200	128734	128462	59521	51640
L_LOin_b	ERR1906857	108906	108115	107927	51067	41901
L_LOin_c	ERR1906858	110955	109660	109428	51843	45147
S_Kal_a	ERR1906859	95387	94836	94615	41880	37249
S_Kal_b	ERR1906860	100113	99321	99087	44317	41666
S_Kal_c	ERR1906861	100822	100293	100012	44011	40730
R_ARDelta_a	ERR1906862	99760	99076	98734	34367	33110
R_ARDelta_b	ERR1906863	120241	119374	118970	42085	39841
R_ARDelta_c	ERR1906864	94144	93618	93131	32188	30936
R_AR_a	ERR1906865	97520	96530	96379	46297	41914
R_AR_b	ERR1906866	157246	156223	155950	76214	70533
R_AR_c	ERR1906867	167487	167016	166742	80555	73018
R_ARO_a	ERR1906868	124486	123858	123648	54574	52558
R_ARO_b	ERR1906869	148083	147656	147402	68979	62077
R_ARO_c	ERR1906870	142040	141718	141455	65768	58984
SUM		4355050	4325440	4315704	1036216	867332

In all pre-processing steps until the merging of the 2 reads of each sample, there are two .fastq files for each sample, as reads are derived from paired-end Illumina sequencing. Apparently, in case of Amvrakikos gulf and the 16S marker gene dataset, at first, there were 4.355.050 reads but only 867.332 remained (20%) after the pre-processing and proceeded to the clustering step. Respectively, in the COI dataset, there were initially 47.551.240 reads, and finally only 1.288.237 remained (3%).

The folder called “6.linearized_files” contains the sequences that remained after they were treated properly to form a single .fasta (“teliko_all_samples.fasta”). That is the file P.E.M.A. will use from this point onwards for the clustering and taxonomy assignment steps.

3.2 Step 7: Clustering

For the case of 16S marker gene and USEARCH algorithm, 5 files are produced while in the same directory there is also a folder with the output of the next step of P.E.M.A. (taxonomy assignment).

The sequences that were defined as MOTUs (Molecular Operational Taxonomic Unit) can be found in the “16S_all_samples.otus.fa” file.

```
haris@zorba:~/metabar_pipeline/results/16S_analysis_for_statistics/7.gene_dependent/gene_16S$ ll
total 693471
-rw-r--r-- 1 haris users 413160451 Αύγ  30 16:27 16S_all_samples.fasta
-rw-r--r-- 1 haris users   2076042 Αύγ  30 16:27 16S_all_samples.otus.fa
-rw-r--r-- 1 haris users   3097712 Αύγ  30 16:27 16S.all_samples.uparse.txt
-rw-r--r-- 1 haris users  15024219 Αύγ  30 16:27 16S_map.txt
-rw-r--r-- 1 haris users 974386515 Αύγ  30 16:27 16S_mysilvamod128.xml
-rw-r--r-- 1 haris users    206509 Αύγ  30 16:27 16S_otutab.txt
drwxr-xr-x 2 haris users        20 Αύγ  30 16:27 16S_taxon_assign
```

Figure 17: The output files of clustering step in the case of 16S and Swarm clustering algorithm

Another file is created by USEARCH, "16S.all_samples.uparse.txt" (Figure 18). In this, there are five or six fields. In the first one, the query label is indicated. The second one, can take one of these 4 values: "otu", "match", "perfect" and "chimera". These correspond to the 4 states that could characterize this sequence: being a new OTU, to be assigned to another already existing OTU (>= 97% match), matching perfectly with one or being a chimeric sequence. In the third column, the percent identity between the query sequence and the top hit in the reference database is recorded. If there is no hit, this field is set to "*".

```
haris@zorba:~/metabar_pipeline/results/16S_analysis_for_statistics/7.gene_dependent/gene_16S$ head *uparse.txt
ERR1906855:293;size=41  OTU     *
ERR1906863:740;size=38  OTU     dqt=118;
ERR1906853:800;size=38  perfect top=Otu1(100.0%);
ERR1906856:5987;size=35 OTU     dqt=128;
ERR1906855:5405;size=35 match   dqt=2;top=Otu1(99.6%);
ERR1906863:869;size=34  match   dqt=1;top=Otu2(99.8%);
ERR1906855:425;size=33  match   dqt=1;top=Otu1(99.8%);
ERR1906855:2545;size=33 OTU     dqt=58;
ERR1906855:2535;size=32 match   dqt=2;top=Otu1(99.6%);
ERR1906863:677;size=29  match   dqt=1;top=Otu2(99.8%);
```

Figure 18: The output of the uparse.txt file. Here 3 of the 4 possible values for a query sequence are shown.

Making use of Amvrakikos dataset, 4.457 OTUs were found. The OTU-table that includes the OTUs found and the number of the copies observed in each sample, lies in the file "16S_otutab.txt" (Figure 19). Obviously, in this OTU-table there are no taxonomies. That is the main task of P.E.M.A's next step.

#OTU ID	ERR1906856		ERR1906863		ERR1906855		ERR1906853		ERR1906857		ERR1906859		ERR1906858		ERR1906870		ERR1906854		ERR1906867 $
Otu3	1013	3	85	84	259	427	575	89	45	16	309	21	0	0	136	155	110	105	
Otu2	0	1453	1	5	0	0	0	0	0	0	0	0	458	480	0	0	0	0	
Otu1	2014	212	3888	3667	2661	22	1904	4	1853	105	109	20	134	108	4	32	6	7	
Otu5	833	12	720	449	1021	6	665	0	360	0	0	0	6	3	0	14	5	0	
Otu6	0	0	1	0	0	1148	1	0	0	0	0	0	0	0	0	681	628	0	
Otu4	673	107	1363	1205	982	0	980	8	539	37	24	31	63	73	6	2	0	1	
Otu7	468	152	1094	214	735	12	510	0	477	2	1	0	72	220	2	51	35	0	
Otu258	711	93	1039	1607	1049	4	710	1	834	21	12	7	46	42	2	2	1	0	

Figure 19: Part of the OTU-table for the case of 16S marker gene.

In case of COI marker gene and SWARM clustering algorithm, the output is similar. The file "SWARM_otu_no_chimera.fasta" contains all the MOTUs found. As SWARM

does the clustering and then the chimera removal takes place, in this file only the true MOTU sequences are included. Contrary, MOTU representatives are included in the "SWARM_final_OTU_representative.fasta" .

SWARM also produces two files ".stats" (Figure 20) and ".swarms". The first one is a tab-separated table with one MOTU per row and 8 columns of information. The number of unique amplicons in the MOTU, the total abundance of amplicons in it, the identifier of the initial seed, the abundance of that initial seed and the number of amplicons with abundance 1 stand for the first 6 columns of ".stats" file, respectively. The last two columns represent the maximum number of iterations before the MOTU reached its natural limit and the cumulative number of steps along the path joining the seed and the furthermost amplicon in the MOTU. Furthermore, in case that parameter*d* equals to 1, all first five columns are modified, as a second clustering pass is performing to reduce the number of small MOTUs, but not the last two.

```
haris@zorba:~/metabar_pipeline/results/final_COI_d_2/7.gene_dependent/gene_COI/SWARM$ head SWARM.stats
13399   28135   ERR1308238:23861        33      8951    7       14
4880    7968    ERR1308298:196911       15      3260    5       9
775     2921    ERR1308238:54541        14      38      4       8
14115   16975   ERR1308237:12561        13      12112   5       10
7496    11008   ERR1308252:16185        13      5656    8       16
1       12      ERR1308238:19662        12      0       0       0
5686    8756    ERR1308242:25110        12      4003    7       14
4       41      ERR1308252:15252        12      0       2       4
9113    11955   ERR1308206:39955        10      7232    5       10
1       10      ERR1308302:42660        10      0       0       0
```

Figure 20: A typical image of the .stats file.

The MOTUs are written in the ".swarms" file. In fact, each line of this file, contains as much MOTUs as it is mentioned in the first column of the ".stats" file.

```
ERR1308246:27793_3 ERR1308246:3329_2 ERR1308246:114694_1 ERR1308246:125546_1 ERR1308246:137283_1 ERR1308246:139403_1 ERR1308246:146805_1 ERR13
ERR1308247:102932_3 ERR1308247:138521_2 ERR1308247:74995_2 ERR1308247:24802_2
ERR1308247:112462_3 ERR1308247:87638_2
ERR1308247:118017_3 ERR1308247:7016_3
ERR1308247:123934_3 ERR1308247:29943_3
ERR1308247:151865_3 ERR1308247:103522_1 ERR1308247:152374_1 ERR1308247:182166_1 ERR1308247:188186_1 ERR1308247:29805_1 ERR1308247:30187_1 ERR1
ERR1308247:15508_3
ERR1308247:17634_3 ERR1308247:35446_3
ERR1308247:18063_3
ERR1308247:19020_3
```

Figure 21: A typical image of the .swarms file where it is shown that amplicons from different samples are clustered in the same MOTU

As SWARM was performed for 4 different values of parameter d, it returned 885759, 839596, 745507and 656283 MOTUs, respectively. However, after the chimera removal, 885752, 839596, 745506 and 655863 remained respectively.

3.3 Step 8: Taxonomy assignment

After the clustering step, the assignment of the MOTUs found was achieved through three different approaches.

3.3.1 16S

For the assignment of the OTUs arised from 16S marker gene, P.E.M.A. has two alternatives: 1) an alignment-based approach using CREST and the program LCAClassifier and 2) a phylogeny-based approach that uses a reference tree of 1000 'consensus taxa'. SILVA is the database used in both approaches.

At first, the alignment-based approach is performed. LCAClassifier creates a folder called "16S_taxon_assign" in which its output is stored. Among its derivatives are the files "Relative_Abundance.tsv", "All_Assignments.tsv", "Richness.tsv" and "16S_otutab.txt" (Figures 22-26 respectively). The "Relative_Abundance.tsv" file (Figure 22) contains relative abundance data across the dataset, which are normalised to the total number of assigned reads.

GNU nano 2.7.4 File: Relative_Abundance.tsv

Rank	Taxonpath	Taxon	ERR1906855	ERR1906853	ERR1906863	ERR1906856	ERR1906857	ERR1906859	ERR1906858	ERR19068
root	root	root	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
meta	Main genome	Main genome	0.994353118334	0.994122097422	0.997494375128	0.968239515827	0.989408699592	0.898782999438	0.976025472935	
meta	Chloroplast	Chloroplast	0.00564688166646	0.00587790257813	0.00250562487216	0.0317604841733	0.0105913004078	0.099082		
domain	Main genome;Bacteria	Bacteria	0.984899820136	0.969912122447	0.852935160565	0.893765732955	0.977826234708	0.889758472196	0.955172	
domain	Main genome;Archaea	Archaea	0.00945329819718	0.0242099749753	0.144559214563	0.0744737828718	0.0115824648845	0.00902452724209		
domain	Chloroplast;Eukaryota (Chloroplast)	Eukaryota (Chloroplast)	0.00564688166646	0.00587790257813	0.00250562487216	0.031760				
superkingdom	Main genome;Bacteria;Bacteria (superkingdom)	Bacteria (superkingdom)	0.984899820136	0.969912122447	0.852935160565	0.893765732955				
superkingdom	Main genome;Archaea;Archaea (superkingdom)	Archaea (superkingdom)	0.00945329819718	0.0242099749753	0.144559214563	0.074473				
superkingdom	Chloroplast;Eukaryota (Chloroplast);Ptilocladiopsis (Chloroplast)	Ptilocladiopsis (Chloroplast)	0.0	0.0	0.0	0.0				
superkingdom	Chloroplast;Eukaryota (Chloroplast);SAR (Chloroplast)	SAR (Chloroplast)	0.0	0.0	0.0	0.0	0.0	0.00190975472758		
kingdom	Main genome;Bacteria;Bacteria (superkingdom);FCB group	FCB group	0.389551177479	0.614240819415	0.301646553487	0.324888865085	0.408274			
kingdom	Main genome;Bacteria;Bacteria (superkingdom);Proteobacteria (superphylum)	Proteobacteria (superphylum)	0.370958296733	0.185474015015						
kingdom	Main genome;Bacteria;Bacteria (superkingdom);Terrabacteria	Terrabacteria	0.104049023299	0.0896525635803	0.133871957456	0.134433078035				

Figure 22: The table with the relative abundances of the OTUs per sample

The number of assignments at each taxonomic rank are provided in "All_Assignments.tsv" (Figure 23). Assignments to the taxon node itself are counted only and not to child taxa at lower ranks. For each taxon, the full taxonomic path from root to the taxon itself is also provided.

GNU nano 2.7.4 File: All_Assignments.tsv

Rank	Taxonpath	Taxon	ERR1906855	ERR1906853	ERR1906863	ERR1906856	ERR1906857	ERR1906859	ERR1906858	ERR1906870
root	root	root	39769	29198	14229	30012	29705	20320	26174	32404	34247	41492	36090	24807	19839	12254	11183	32890	17299
meta	Main genome	Main genome	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
meta	Chloroplast	Chloroplast	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
domain	Main genome;Bacteria	Bacteria	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
domain	Main genome;Archaea	Archaea	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
domain	Chloroplast;Eukaryota (Chloroplast)	Eukaryota (Chloroplast)	270	202	49	1186	374	2637	768	1876	304	254	1429						
superkingdom	Main genome;Bacteria;Bacteria (superkingdom)	Bacteria (superkingdom)	46398	33072	16477	32057	33539	22438	29742	39499	36962								
superkingdom	Main genome;Archaea;Archaea (superkingdom)	Archaea (superkingdom)	412	757	2702	2452	346	208	526	457	4810								
superkingdom	Chloroplast;Eukaryota (Chloroplast);Ptilocladiopsis (Chloroplast)	Ptilocladiopsis (Chloroplast)	0	0	0	0	0	0	0	0	0								
superkingdom	Chloroplast;Eukaryota (Chloroplast);SAR (Chloroplast)	SAR (Chloroplast)	0	0	0	0	0	0	0	0	0								
kingdom	Main genome;Bacteria;Bacteria (superkingdom);FCB group	FCB group	0	0	0	0	0	0	0	0	0								
kingdom	Main genome;Bacteria;Bacteria (superkingdom);Proteobacteria (superphylum)	Proteobacteria (superphylum)	0	0	0	0	0	0	0	0	0								
kingdom	Main genome;Bacteria;Bacteria (superkingdom);Terrabacteria	Terrabacteria	0	0	0	0	0	0	0	0	0								

Figure 23: The table with the taxonomic assignments of the OTUs per sample

In “All_Cumulative.tsv” file (Figure 24), cumulative counts for the number of assignments at each taxonomic rank are listed. Contrary to “All_Assignments.tsv”, here assignments to child taxa are counted too.

GNU nano 2.7.4 File: All_Cumulative.tsv

Rank	Taxonpath	Taxon	ERR1906855	ERR1906853	ERR1906863	ERR1906856	ERR1906857	ERR1906859	ERR1906858	ERR1906870
root	root	root	47814 34366 19556	37342 35312	26705 32034	41908 43261	48432 40306	27478 26131	16320 15360	43643 24220
meta	Main genome	Main genome	47544 34164 19507	36156 34938	24002 31266	40032 42957	48178 38877	27381 24746	16310 15341	
meta	Chloroplast	Chloroplast	270 202 49	1186 374	2646 768	1876 304	254 1429	97 1372	10 19	
domain	Main genome;Bacteria	Bacteria	47092 33332 16680	33375 34529	23761 30598	39574 37645	48117 38582	27180 24418	13159	
domain	Main genome;Archaea	Archaea 452	832 2827	2781 409	241 668	458 5312	61 295	201 328	3151 2272	
domain	Chloroplast;Eukaryota (Chloroplast)	Eukaryota (Chloroplast) 270	202 49	1186 374	2646 768	1876 304	254 1429			
superkingdom	Main genome;Bacteria;Bacteria (superkingdom)	Bacteria (superkingdom) 47092	33332 16680	33375 34529	23761 30598	39574 37645				
superkingdom	Main genome;Archaea;Archaea (superkingdom)	Archaea (superkingdom) 452	832 2827	2781 409	241 668	458 5312				
superkingdom	Chloroplast;Eukaryota (Chloroplast);Ptilocladiopsis (Chloroplast)	Ptilocladiopsis (Chloroplast) 0	0 0	0 0	0 0					
superkingdom	Chloroplast;Eukaryota (Chloroplast);SAR (Chloroplast)	SAR (Chloroplast) 0	0 0	0 0	51 0	0				
kingdom	Main genome;Bacteria;Bacteria (superkingdom);FCB group	FCB group 18626	21109 5899	12132 14417	4683 12664	12157 11729	13968			
kingdom	Main genome;Bacteria;Bacteria (superkingdom);Proteobacteria (superphylum)	Proteobacteria (superphylum) 17737	6374 4070	11371 12211						
kingdom	Main genome;Bacteria;Bacteria (superkingdom);Terrabacteria	Terrabacteria 4975	3081 2618	5020 3558	1999 3525	6900 6227				

Figure 24: The table with the total number of the amplicons assigned to each OTU per sample

Total count of OTUs for each taxon as well as their number can be found in “Richness.tsv” (Figure 25).

GNU nano 2.7.4 File: Richness.tsv

Rank	Taxonpath	Taxon	ERR1906855	ERR1906853	ERR1906863	ERR1906856	ERR1906857	ERR1906859	ERR1906858	ERR1906870
root	root	root	1363 1172 1368	1513 1279	1075 1351	1598 1581	1871 2021	1587 1245	1179 1223	1577 1226
meta	Main genome	Main genome	1353 1163 1361	1504 1270	1051 1343	1586 1571	1859 2011	1580 1220	1174 1219	
meta	Chloroplast	Chloroplast	10 9 7	9 9	23 8	12 10	12 10	7 24	5 4	
domain	Main genome;Bacteria	Bacteria	1267 1068 1180	1336 1170	1015 1211	1542 1385	1841 1973	1552 1165	985	
domain	Main genome;Archaea	Archaea 86	95 181 168	100 36	132 44	186 18	38 28	55 189	174	
domain	Chloroplast;Eukaryota (Chloroplast)	Eukaryota (Chloroplast) 10	9 7	9 9	23 8	12 10	12 10			
superkingdom	Main genome;Bacteria;Bacteria (superkingdom)	Bacteria (superkingdom) 1267	1068 1180	1336 1170	1015 1211	1542 1385				
superkingdom	Main genome;Archaea;Archaea (superkingdom)	Archaea (superkingdom) 86	95 181	168 100	36 132	44 186				
superkingdom	Chloroplast;Eukaryota (Chloroplast);Ptilocladiopsis (Chloroplast)	Ptilocladiopsis (Chloroplast) 0	0 0	0 0	0 0					
superkingdom	Chloroplast;Eukaryota (Chloroplast);SAR (Chloroplast)	SAR (Chloroplast) 0	0 0	0 1	0 0					
kingdom	Main genome;Bacteria;Bacteria (superkingdom);FCB group	FCB group 420	412 301	400 376	219 377	426 407	467			
kingdom	Main genome;Bacteria;Bacteria (superkingdom);Proteobacteria (superphylum)	Proteobacteria (superphylum) 317	231 297	297 277						
kingdom	Main genome;Bacteria;Bacteria (superkingdom);Terrabacteria	Terrabacteria 174	142 178	189 153	102 149	268 210				

Figure 25: The table with the total number of OTUs found per taxonomic group.

Finally, “16S_otutab.txt” (Figure 26) is the OTU-table that P.E.M.A. ends up with. The OTU-table contains all information about how OTUs are distributed, and hence it contains the taxonomic composition across each sample of the dataset.

OTU	ERR1906855	ERR1906853	ERR1906863	ERR1906856	ERR1906857	ERR1906859	ERR1906858	ERR1906870	ERR1906854	ERR1906867	E
RR1906866	ERR1906865	ERR1906861	ERR1906862	ERR1906864	ERR1906869	ERR1906860	ERR1906868	classification			
Otu4056	8 4	0 0	8 3	6 0	3 0	0 0	0 0	0 2	0	Main genome;	
Bacteria;Bacteria (superkingdom);Proteobacteria (superphylum);Proteobacteria;Gammaproteobacteria;Vibrionales;Vibrionaceae;Vibrio											
Otu4057	23 5	1 6	8 0	13 0	2 0	0 0	0 3	0 0	0	Main genome;	
Bacteria;Bacteria (superkingdom);FCB group;Caldithrix phylum incertae sedis;Caldithrix class incertae sedis;Caldithrix order incertae sedis;Caldithrix family incert											
ae sedis;Unknown Caldithrix family incertae sedis genus 4											
Otu4054	0 0	0 1	2 0	12 0	0 0	0 0	0 0	0 0	0	Main genome;	
Bacteria;Bacteria (superkingdom);Fusobacteria (superphylum);Fusobacteria;Fusobacteriia;Fusobacteriales											
Otu4055	0 1	0 4	9 0	9 0	0 1	0 0	0 0	0 0	0	Main genome;	
Bacteria;Bacteria (superkingdom);Proteobacteria (superphylum);Proteobacteria;Deltaproteobacteria;Myxococcales;Unknown Myxococcales family 16											
Otu4052	1 3	2 2	1 0	5 0	1 0	0 0	1 2	0 0	3	0	Main genome;
Bacteria;Bacteria (superkingdom);Proteobacteria (superphylum);Proteobacteria;Deltaproteobacteria;Myxococcales;VHS-B4-70											
Otu4053	0 0	0 1	4 0	11 0	0 0	0 13	0 0	0 0	0	Main genome;	
Bacteria;Bacteria (superkingdom);CPR;Ca. Parcubacteria;Candidatus Falkowbacteria											
Otu4050	5 3	1 5	2 0	9 0	3 0	0 0	0 0	0 0	0	Main genome;	
Bacteria;Bacteria (superkingdom);FCB group;Bacteroidetes;Sphingobacteriia;Sphingobacteriales;Unknown Sphingobacteriales family 25											
Otu4051	0 0	0 5	6 0	14 0	0 0	1 0	8 0	1 0	9	0	Main genome;
Bacteria;Bacteria (superkingdom);Proteobacteria (superphylum);Proteobacteria;Deltaproteobacteria;Desulfobacterales;Desulfobulbaceae;Desulfobulbus											
Otu4058	2 6	0 15	12 12	9 3	4 11	8 2	4 0	0 0	0	2	Main genome;

Figure 26: The OTU-table after the taxonomic assignment, where taxonomy is placed in the last column

In case of Amvrakikos dataset, some of the OTUs found were present in all samples while others only either in the marine or the lagoonal or the riverine samples.

After getting the OTU-table and making use of some bash and "awk" commands, a series of hypotheses about which species were expected in each of the three environments (river, lagoon, sea) were tested. In the figure below (Figure 27), the OTUs assigned to the class of Betaproteobacteria are shown and apparently, while they are commonly present in the riverine environment, they are totally absent from marine and lagoonal samples.

GNU nano 2.7.4 File: Betaproteobacteria.txt

OTU	ERR1906853	ERR1906854	ERR1906855	ERR1906856	ERR1906857	ERR1906858	ERR1906859	ERR1906860	ERR1906861	ERR1906862 $
Otu1327	0	0	0	0	0	0	21	39	2	7	5	5	Betaproteob$
Otu2660	0	0	0	0	0	0	0	0	0	5	16	11	Betaproteob$
Otu794	0	0	0	0	0	0	0	0	1	42	18	41	Betaproteob$
Otu3044	0	0	0	0	0	0	16	15	11	5	7	10	Betaproteob$
Otu3108	0	0	0	0	0	0	9	7	15	0	0	0	Betaproteob$
Otu3213	0	0	0	0	0	0	0	4	5	10	16	3	Betaproteob$
Otu2890	0	0	0	0	0	0	0	1	0	9	15	19	Betaproteob$
Otu2892	0	0	0	0	0	0	1	11	5	29	26	34	Betaproteob$
Otu2893	0	0	0	0	0	0	0	0	0	3	6	15	Betaproteob$
Otu786	0	0	0	0	0	0	7	20	29	26	96	116	Betaproteob$
Otu1688	0	0	0	0	0	0	4	26	0	0	0	0	Betaproteob$
Otu3171	0	0	0	0	0	0	0	1	3	0	0	0	Betaproteob$
Otu3218	0	0	0	0	0	0	0	1	4	12	14	10	Betaproteob$
Otu774	0	0	0	0	0	0	0	3	5	32	64	58	Betaproteob$
Otu772	0	0	0	0	0	0	0	1	0	53	47	53	Betaproteob$
Otu3303	0	0	0	0	0	0	0	6	12	25	17	12	Betaproteob$
Otu3272	0	0	0	0	0	0	1	0	4	6	3	10	Betaproteob$
Otu1318	0	0	0	0	0	0	2	26	0	4	20	8	Betaproteob$
Otu365	0	0	0	0	0	0	0	5	1	45	197	192	Betaproteob$
Otu364	0	0	0	0	0	0	7	15	13	39	175	218	Betaproteob$
Otu367	0	0	0	0	0	0	1	1	0	48	178	155	Betaproteob$
Otu488	0	0	0	0	0	1	0	1	125	103	55	9	8	7	Betaproteob$
Otu1647	0	0	0	0	0	0	0	0	0	1	8	0	0	0	Betaproteob$
Otu2915	0	0	0	0	0	0	1	2	1	15	33	42	Betaproteob$

Figure 27: Example of the OTU table showing the OTUs assigned to Betaproteobacteria.

The phylogeny-based approach is then performed and another folder ("16S_taxon_assign_phylogeny_assignment") is been created by P.E.M.A. During this approach, a multiple sequence alignment (MSA) is constructed by PaPaRa algorithm. This MSA is supposed to be made with both the reference sequences and the queries; the final file is supposed to contain only the alignment of the query sequences as it ensued. However, due to a bug on PaPaRa's code the reference sequences are removed from the final MSA by P.E.M.A. which subsequently executes the "convertPhylipToFasta.sh" manually written program, to convert this final MSA from phylip (.phy) to Fasta (.fasta) format.

Finally, EPA-ng is performed using the MSA file ("papara_alignment.fasta") along with the reference MSA ("raxml_easy_right_refmsa.raxml.reduced.phy.fasta") and the reference tree ("raxml_easy_right_refmsa.raxml.bestTree"). At last, in the folder for the phylogeny-based assignment, two output files are included: the "epa_info.log" which includes all parameters as they were set in EPA-ng and the "epa_result.jplace" file. The .jplace file is the final output of this approach as it is the one that can be used as an input to a series of different tools (e.g. iTOL) in order to visualize the assignments of the OTUs found to the reference tree of 1000 taxa. In the figures below (Figures 28-29), a small segment of the tree (Figure 28) and the information that it is included in each node, are shown. The more OTUs placed to a specific taxon, the larger the circles become.

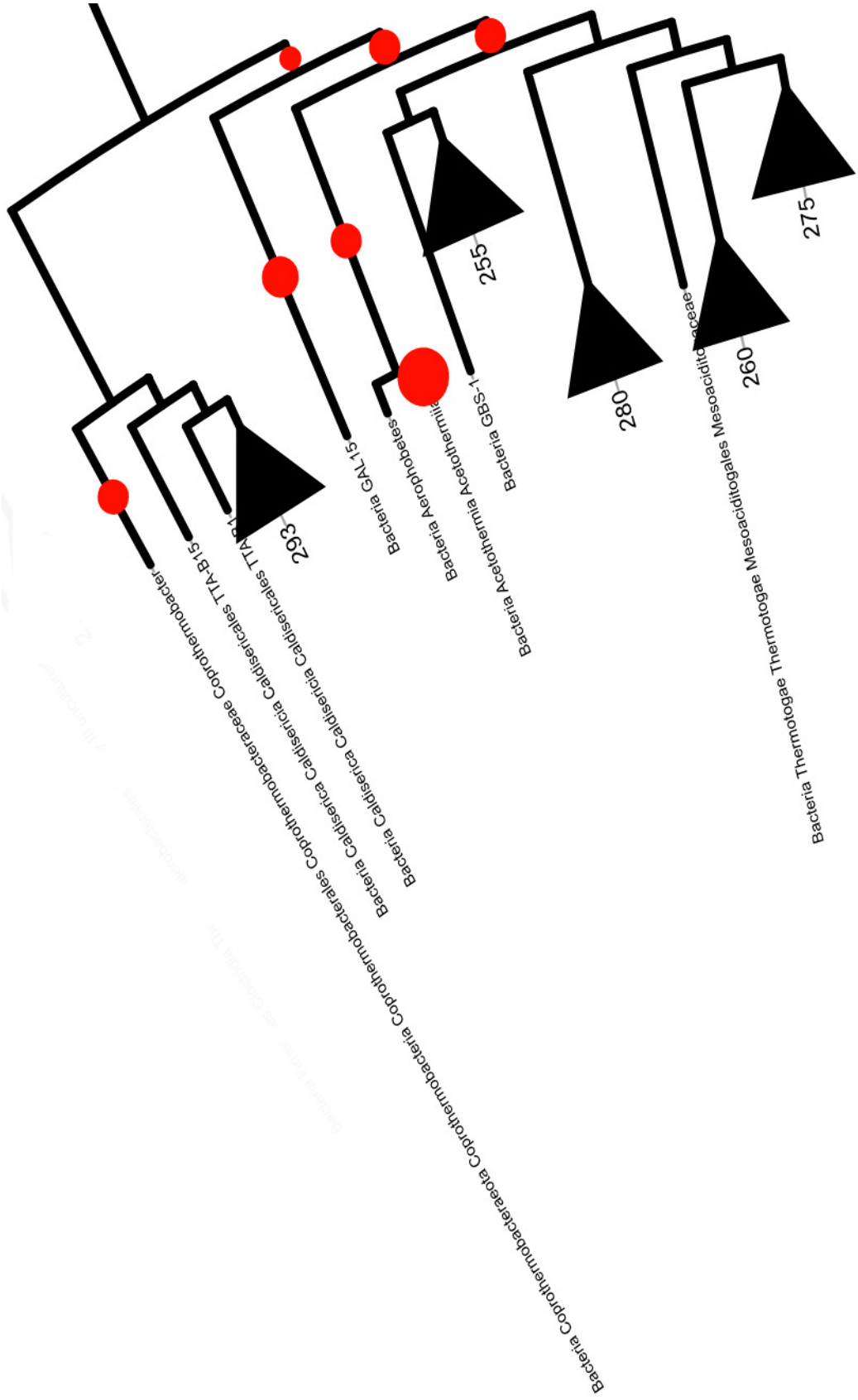


Figure 28: A part of the phylogeny reference tree after the taxonomy assignment of the OTUs found from the Amvrakikos dataset. Red circles: taxonomic placements.

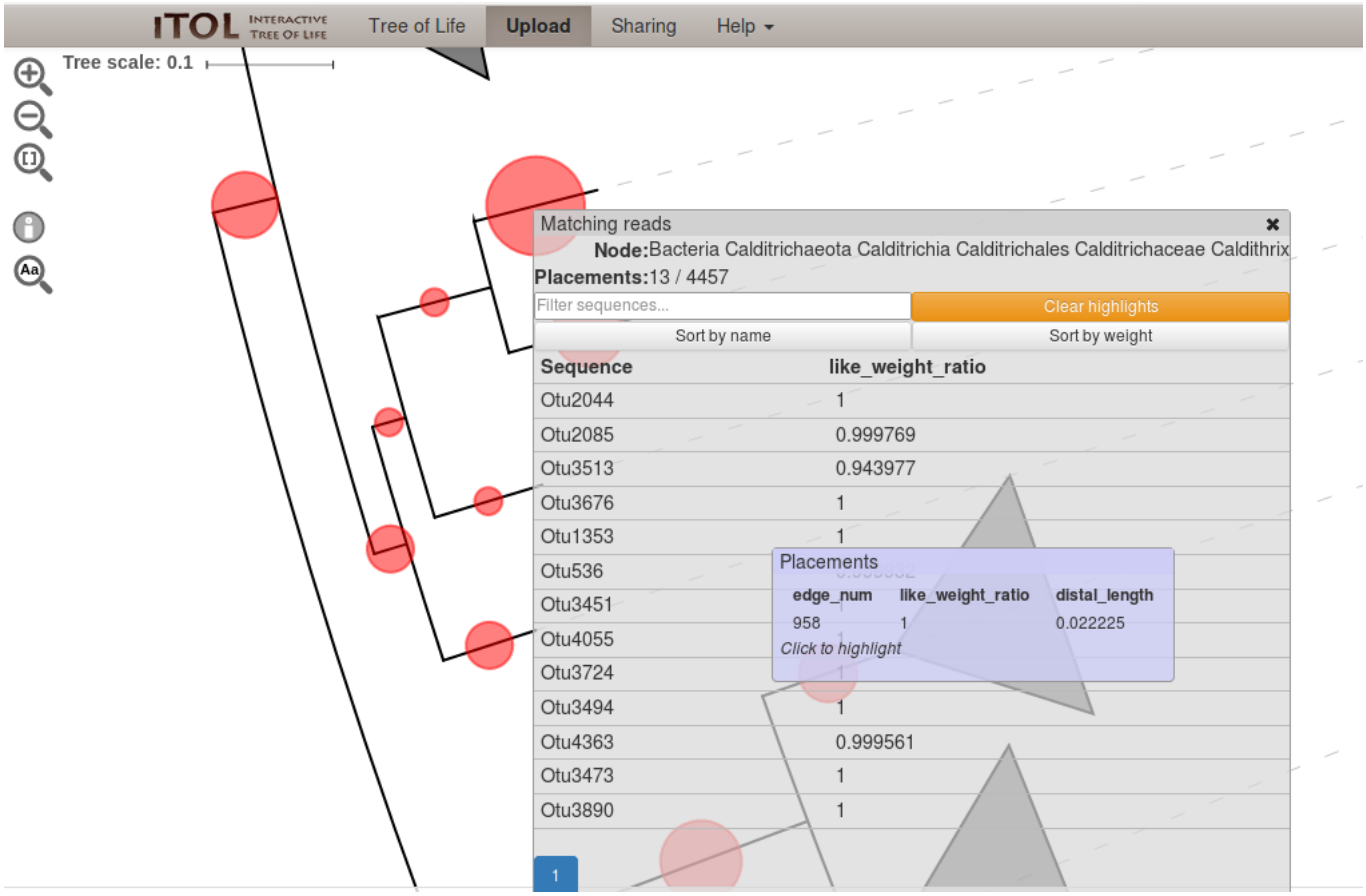


Figure 29: Example of the OTU table showing the OTUs assigned to Betaproteobacteria.

3.3.2 COI

In case of COI marker gene, P.E.M.A. performs an alignment-based approach for the taxonomy assignment of the retrieved MOTUs. The equivalent of CREST - LCAClassifier this time is the RDPClassifier and instead of SILVA, the MIDORI database is used.

LCAClassifier 's output can be found in the "SWARM" folder that P.E.M.A had already created and it is called "tax_assign_swarm_COI.txt" (Figure 30). In this file, all MOTUs that SWARM ended up with, are assigned to species level using sequences deposited in the MIDORI database. Each MOTU has a taxonomy and next to each taxon level, the percentage of similarity that the MOTU belongs to that taxon is reported.

```
 GNU nano 2.7.4                                File: tax_assign_swarm_COI.txt

ERR1308296:19748;size=26455     Eukaryota       superkingdom    1.0     Arthropoda      phylum  0.55    Insecta class   0.49    Hymenoptera     order   0.09    Trigonalidae    family
ERR1308238:23861;size=22231     Eukaryota       superkingdom    1.0     Arthropoda      phylum  0.96    Insecta class   0.95    Diptera order   0.71    Drosophilidae   family  0.32
ERR1308209:37909;size=9761      Eukaryota       superkingdom    1.0     Entoprocta      phylum  0.02    Loxosomatidae   family  0.02    Loxosomella     genus   0.02    Loxosomella a
ERR1308206:107928;size=9376     Eukaryota       superkingdom    1.0     Porifera        phylum  0.15    Homoscleromorpha                class   0.09    Homosclerophorida       order   0.09
ERR1308252:16185;size=9328      Eukaryota       superkingdom    1.0     Arthropoda      phylum  0.99    Insecta class   0.96    Diptera order   0.73    Chironomidae    family  0.2
ERR1308298:196911;size=7692     Eukaryota       superkingdom    1.0     Arthropoda      phylum  0.4     Arachnida       class   0.14    Araneae order   0.09    Cybaeidae       family
ERR1308247:24925;size=7508      Eukaryota       superkingdom    1.0     Arthropoda      phylum  0.96    Insecta class   0.91    Diptera order   0.78    Muscidae        family  0.23
```

Figure 30: The output of LCAClassifier for the case of COI marker gene after clustering MOTUs with Swarm algorithm.

As a great number of MOTUs that are returned by Swarm are singletons, P.E.M.A.

automatically removes them. In the chosen dataset, 885.752 MOTUs were found with the 877.728 of them being singletons. Hence, the rest of the MOTUs (8024 in our case) are the only ones that were assigned to a taxonomy.

As LCAClassifier has to assign MOTUs exclusively to species level, a great number of assignments may have a very low percentage of similarity with the taxon assigned to, even less than 10%. Hence, such assignments have to be removed either completely or partially, i.e. by keeping the lowest taxonomic resolution which can even be at the family level. For this reason, the final OTU-table for the case of the COI marker gene, is not made by the "tax_assign_swarm_COI.txt" file, but after a pre-process on that, during which only the assignments that have more than 97% similarity to the MIDORI taxa, are kept.

In the table below (Table 6), the number of MOTUs found with the different values of parameter *d*, the hit taxonomies, and the numbers of the unique assigned species are recorded.

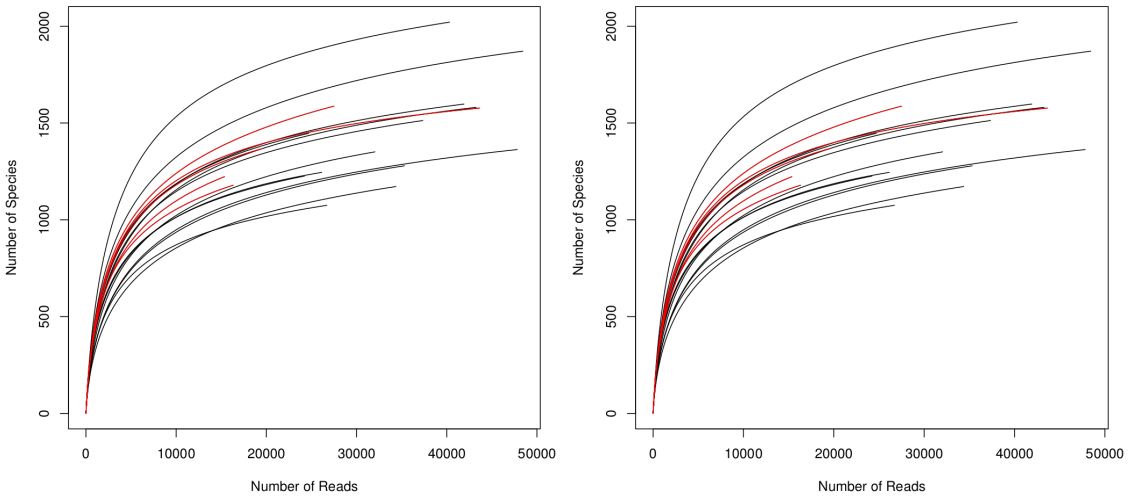
Table 6: P.E.M.A.'s outputs for different values of *d* parameter of Swarm.

	d=1	*d*=2	*d*=3	*d*=4	*d*=10
MOTUs	885759	839596	745507	656283	194023
MOTUs after chimera removal	885752	839596	745506	655863	193819
Taxonomies that were hit	77	116	117	127	155
Assigned species	56	81	83	90	114

3.4 Step 9: Rhea: biodiversity of Amvrakikos gulf

For the case of 16S marker gene and after building a phylogenetic tree for the P.E.M.A. retrieved OTUs, the Rhea set of scripts was used for the next steps of the analysis of Amvrakikos' dataset. After the normalization step of the produced OTU-table, two plots were generated by Rhea, estimating the sufficiency of sequencing depth, in which rarefaction curves of all samples (Figure 31a) and the most undersequenced samples (Figure 31b) are shown, respectively.

Alpha- and beta-diversity were also retrieved (Figures 32 and 33 respectively), using the mapping file of *Pavloudi et al.* (2017).

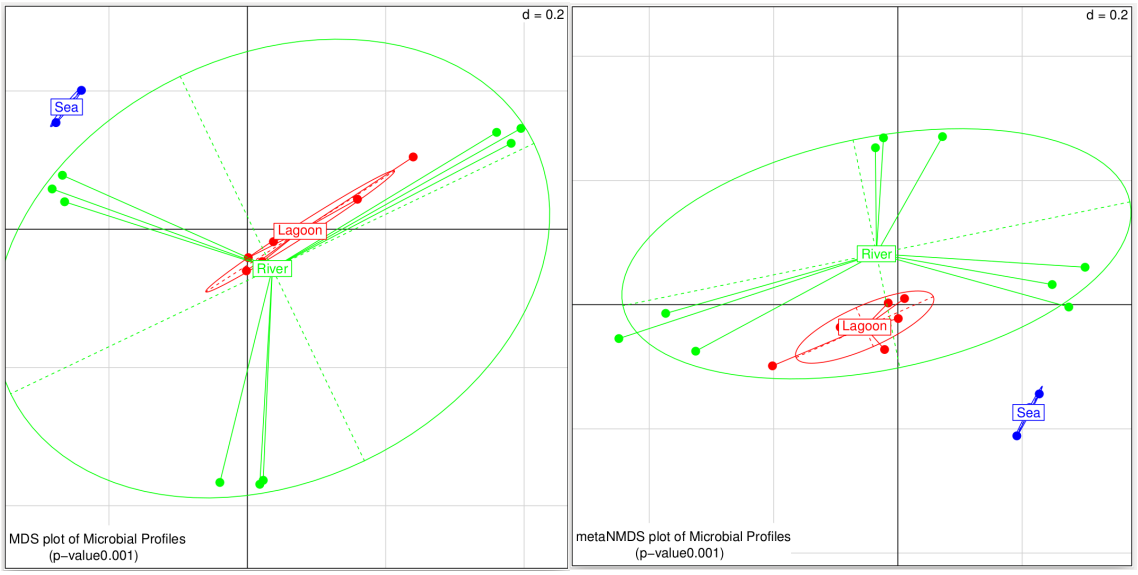


(a) (b)

Figure 31: Rarefaction curves of: (a) all samples, (b) the 5 most undersequenced samples, as returned by Rhea.

```
haris@zorba:~/metabar_pipeline/results/16S_analysis_for_statistics/Rhea_otus/2.Alpha-Diversity$ head alpha-diversity.tab
          Richness         Shannon Shannon.effective         Simpson Simpson.effective       Evenness
ERR1906853      922   5.31122645551036          202.6  0.0197370454305085          50.67  0.539286201495986
ERR1906854     1343   6.04905815699514         423.71  0.0067030319778175         149.19  0.582130339351273
ERR1906855     1121   5.60688804087368          272.3  0.012828833255335           77.95  0.553462216773799
ERR1906856     1257   5.99915294751033         403.09  0.00806789629449003        123.95  0.582681389386483
ERR1906857     1025   5.58142741024014         265.45  0.0124890156469642          80.07  0.55806415474266
ERR1906858     1091   5.77080199649824         320.79  0.00982515342604255        101.78  0.571851453790122
ERR1906859     1075   5.59061623119161          267.9  0.0115083112306157          86.89  0.555168727567061
ERR1906860     1226 6.0338850679572 417.33  0.0054646847880119          182.99  0.588112676202251
ERR1906861     1245   5.99438218651218         401.17  0.00590204582665736        169.43  0.583001651518452
```

Figure 32: Alpha-diversity metrics of Amvrakikos samples.



(a) (b)

Figure 33: Beta-diversity plots of Amvrakikos samples (a) MDS, (b) metaNMDS, as returned by Rhea.

It is clear that the 3 sampled habitats create their own clusters, significantly different one from another. This can be also seen from the dendrogram output of the Beta-Diversity script (Figure 34), that shows both the distance and the clusters that the individual samples form, based on the Ward's clustering method.

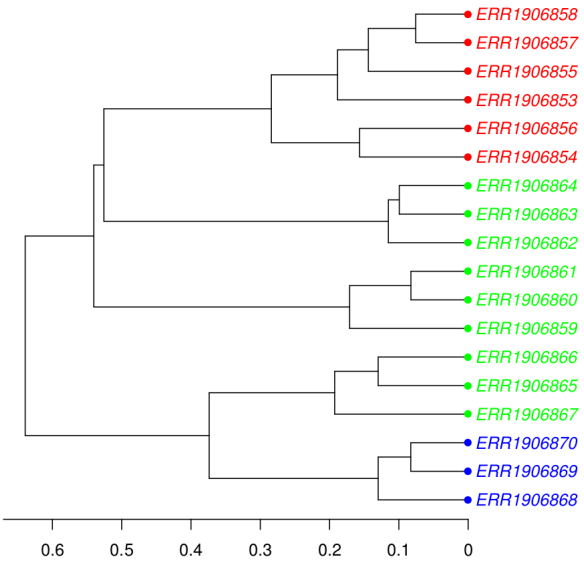


Figure 34: Phylogram of Amvrakikos samples, as returned by Rhea.

3.5 P.E.M.A's statistics

To evaluate P.E.M.A., the execution time was kept in all experiments for both marker genes. Hence in the tables below (Tables 7-8), the execution time needed for each set of parameters for running P.E.M.A. in 6 nodes of "Zorba" cluster is shown.

Table 7: Time needed for P.E.M.A. to be concluded for the different values of *d* parameter of Swarm clustering algorithm.

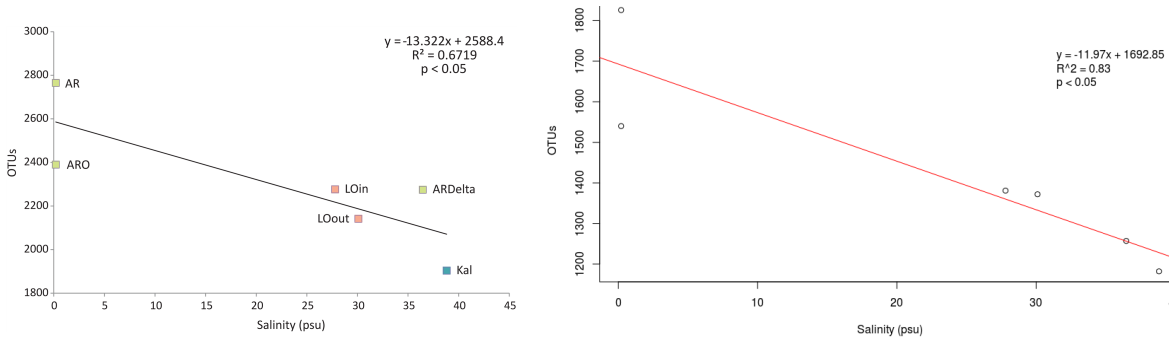
	COI - 5,7GB / 6 nodes
SWARM | *d*=1	11:38:13
SWARM | *d*=2	13:37:57
SWARM | *d*=3	12:49:41
SWARM | *d*=4	11:35:34
SWARM | *d*=10	22:19:23

Table 8: Time needed for P.E.M.A to be concluded for the two different options of taxonomic assignment.

	16S - 843MB / 6 nodes
alignment-based taxonomy assignment	1:25:30
phylogeny-based taxonomy assignment	1:52:15

3.6 Remane's concept validity

As it concerns the investigation of Remane's concept and the comparison with what *Pavloudi et al.* (2017) suggested, the results are shown in the figures below (Figures 35a - 35b), where the number of OTUs found are in the Y axes, while the salinity of each sampling site is in the X axes. It is more than clear that the two studies agree and the models found are statistically significant, with the one that came out from P.E.M.A.'s OTU-table having an even greater value of R square.



(a) (b)

Figure 35: Linear regression models of the number of OTUs (dependent variable) with the salinity values (independent variable, as derived from: (a) Pavloudi et al. 2017, (b) P.E.M.A.

4 Discussion

P.E.M.A. had to be evaluated both from a biological and a computational point of view. Therefore, two published datasets were chosen to test P.E.M.A. and compare its output with the results of their original publications. Furthermore, by being a bioinformatic pipeline, P.E.M.A. needed to be evaluated from such a point of view as well, in an attempt to record its pros and cons.

4.1 The case of 16S marker gene - Amvrakikos' dataset

In the case of 16S marker gene, 4.457 OTUs were found by P.E.M.A, while Pavloudi et. al (2017) came up with 7.050. The different trimming parameters used in each analysis could have caused this variation in the number of OTUs found. However, P.E.M.A. 's output significantly agrees with the one proposed in the research paper. As a matter of fact, this is shown in Rhea's output with the MDS and metaNMDS plots where it became clear that the samples from the three different habitats are distinguished. The same conclusion comes from the phylogram as well.

Moreover, the similarity of the outputs becomes even clearer when the initial question for the validity of Remane's concept is approached. The figures 35a and 35b show similar models that represent the number of species in each salinity level in both cases of P.E.M.A. and the analysis of Pavloudi et al. (2017). Hence, both analyses deny Remane's concept for the case of Amvrakikos gulf.

Furthermore, the alignment-based approach for taxonomic assignment that returned an OTU-table with the number of OTUs of each species found in every sample, provides the researcher with a complete overview of the dataset. In this case, the phylogenetic-based approach does not provide any further information; however, it allows the more

accurate taxonomic assignment of the OTUs and it bears the potential of investigating and focusing with more detail on the OTUs rather than on the samples.

The time needed for P.E.M.A. to run this analysis was relatively short. It took 1 h 25' the case of alignment-based and 1 h 52' in the phylogenetic-based taxonomic assignment. For a metabarcoding pipeline, the required time is a crucial issue as there are numerous of steps that could delay the whole analysis, particularly as the size of the raw data increases.

Moreover, when the building of the phylogenetic tree that comes from the OTUs was performed, almost a day was needed. That is because, it is the number of sites of the sequences that determine how many cores will be used by RAxML. In case of amplicon analysis, the number of sites is hard to overcome 1000 and as 1 core per 500 sites is used by RAxML, 4 cores at most are used at this step. However, as this step is the final of P.E.M.A. and only the execution of Rhea is after that, it does not actually affect P.E.M.A. 's performance.

4.2 The case of COI marker gene - lake dataset

In the case of COI marker gene, the chosen dataset was more complex and with the analysis published, but a few things were comparable between the two analyses. Despite that, 81 species were found by P.E.M.A., while Bista et. al (2017) had 73 OTUs assigned to species level. This number of unique species came up when P.E.M.A. was performed using Swarm algorithm, setting d parameter equal to 2. When the value of d was set equal to 3, the result was similar (83 unique species found) while when d was equal to 1 (as it is suggested by Swarm's developers), only 56 species were identified.

It is perceived that the two analysis differ significantly as both clustering step and taxonomic assignment are implemented otherwise. However, their results about the number of unique species present in the samples, are in agreement to a significant extent. Nevertheless, that is not always the case, as when d parameter of Swarm equals to 1, then the species found are less. This value of d is the suggested one by Swarm's authors [52] hence, the actual number of the unique species, is not certain.

Using Swarm and for$d$ equals to 1, P.E.M.A. was completed in only 11h 38'. When$d$ was set to 2, the time needed was increased, but when it was set to 3 it decreased; this is totally normal for the Swarm algorithm. That is why, Swarm runs as fast as possible (and usually has the best results) for$d$ equals to 1. As d increases, more calculations are needed, and more values need to be kept in memory.

Clustering of the MOTUs it appears to be the most crucial step. The different values of d parameter and the way Swarm is developed, produces the paradox of fewer MOTUs, as d increases, and more unique species found. However, d is not just a clustering threshold for the number of differences allowed between the amplicons. Swarm uses a modified algorithm when d=1 which is considered as a more robust and effective, but

for greater values of d, it could not maintain scaling linearly with increasing amounts of data [52].

On the contrary, CROP algorithm needed a lot of computational time to be performed, as all bayesian clustering algorithms. After a month, the job was quitted. However, it was decided to remain as P.E.M.A.'s option as many studies till now, use it as clustering algorithm.

Contrary to the study of Bista et al. (2017) where BLAST + (megablast) was used for the assignment of the MOTUs to a taxonomy, RDPClassifier and MIDORI database were used by P.E.M.A. In their study, Bista et al. created their own reference COI database by downloading from NCBI GenBank all COI sequences of length about 4100 bp, excluding environmental sequences and higher taxonomic level information [86].

It is known that the 97% 16S rRNA sequence similarity threshold that is commonly used to delineate species is but an approximation. Besides that, the usage of sequence similarity percentage overestimates the evolutionary similarity between pairs of sequences, as it is a non-evolutionary-based distance metric.

4.3 P.E.M.A.: a fast option for metabarcoding analysis

P.E.M.A. is a very fast pipeline for a metabarcoding analysis. This is mainly due to the BDS programming language that allows absolute serialization and lazy processing [88]. This way, even in the case that an error takes place, P.E.M.A. is able to restart from the last checkpoint that was created, before the error occured. In addition, when P.E.M.A. has to perform a "for" loop, then BDS splits that task in all cores available. Then, thanks to the "wait" command that BDS includes, it just stops until everything needed for the next step of the pipeline is completed. However, even if BDS has a lot of advantages, some of the tools included in BDS cannot exploit them. That is why, even if in some steps there are nodes available, P.E.M.A. does not use them.

Part IV

Conclusions: drawbacks & potentials in metabarcoding analysis of eDNA

Metabarcoding is a useful tool to investigate effortlessly and rapidly the taxonomic composition of environmental samples, sometimes on species level or to the lowest level possible. Compared to morphological identification methods, metabarcoding has certain limitations as not all taxa in a sample are detected. However, metabarcoding's strengths, such as objectivity, robustness, timesaving analyses and the ability to detect rare species, are of crucial importance. Especially in biomonitoring assessment, where the expected number of species which come from a wide taxonomic range, is orders of magnitude higher than any approach based on morphological data. However, classical analyses will not be past metabarcoding analysis, since the latter still cannot provide some important ecological and biological information (maturation state, physiological condition, reproductive status, individual sizes or accurate estimation for total biomass of each species) about the different species present in the samples [89].

1 Standardization

One of the major problems in metabarcoding technique, is the non existence of standardized pipelines which, in turn, leads to essential differences among different analyses of the same dataset. It is of essential importance to establish a protocol taking into account the needs of the analysis to be performed (e.g. whether it is species-specific or not, if the samples are eDNA or bulk samples, etc.). This way even the biases that might occur in specific steps, should be fairly reproducible and thus, the comparison among studies would be more reliable.

The non existence of an ideal marker gene that would allow the identification of all taxa, has also to be considered. Given this, specific pipelines have to be standardized for the great groups of taxa, e.g. when microbes are under study, a global pipeline has to be performed. This does not mean that the best algorithms for each step of metabarcoding procedure have already been developed. Contrary, the scientific community should insist in its effort for further improvement of the so-far algorithms. However, some level of customization is always required, according to the question being addressed. A careful consideration of certain preprocessing steps at a minimum, before embarking on further analyses should be global [90].

2 Quantification

As a matter of fact, quantification and more specifically relative abundance as well as species absolute abundance, is one of the most essential challenges in metabarcoding analysis, especially in the case of eDNA. As specimens vary in biomass and are not amplified with the same efficiency because of primer bias, the sequence abundance is substantially skewed among different taxa. For some researchers however, biases between samples should be fairly reproducible. This way the comparison of their relative sequence abundance between samples can be achieved, given that the same laboratory and bioinformatic methods were used for their analysis. This way, semi-quantitative estimates can be retrieved [89].

However, a novel concept about how can the quantification issue in eDNA metabarcoding analysis can be approached came up in the time of this thesis, making use of bulk communities, creating both "test" and "control" sets from the same sample. This approach will be one of our future tasks.

3 Population dynamics and ecological networks

A, yet to come, potential of eDNA metabarcoding analysis is its use for studies of population dynamics as well as for creating ecological networks based on these NGS data. By now, DNA metabarcoding has been used in a series of studies to gain important and accurate information concerning ecology and population dynamics [91].

Ecosystem functioning is a concept built by both species diversity and the interactions that occur between them. Monitoring the response of species interactions to alterations of the environment (either caused by anthropogenic or by natural phenomena) is of crucial importance towards ecosystem conservation. Ecological networks achieve to represent and analyze all these interactions between species. As both the identification and the quantification of the species present in an ecosystem are totally mandatory for such a task, metabarcoding (and eDNA metabarcoding) can be exceptionally auxiliary in deciphering such networks, especially short-term interactions, such as those between predator and prey, as well as cryptic interactions [92] [93].

New opportunities have now arose in understanding both ecological and evolutionary processes in significant large-scale experiments, through the combination of DNA metabarcoding approaches with ecological network analysis. Merging eDNA metabarcoding with ecological network analysis will allow to construct some of the largest, phylogenetically structured species-interaction networks to date. A new era in monitoring biodiversity and ecosystem functioning is rising [94].

References

[1] P. F. Thomsen and E. Willerslev, "Environmental dna–an emerging tool in conservation for monitoring past and present biodiversity," *Biological Conservation*, vol. 183, pp. 4–18, 2015.

[2] S. Shokralla, J. L. Spall, J. F. Gibson, and M. Hajibabaei, "Next-generation sequencing technologies for environmental dna research," *Molecular ecology*, vol. 21, no. 8, pp. 1794–1805, 2012.

[3] C. Mora, D. P. Tittensor, S. Adl, A. G. Simpson, and B. Worm, "How many species are there on earth and in the ocean?," *PLoS biology*, vol. 9, no. 8, p. e1001127, 2011.

[4] J. L. Shaw, L. J. Clarke, S. D. Wedderburn, T. C. Barnes, L. S. Weyrich, and A. Cooper, "Comparison of environmental dna metabarcoding and conventional fish survey methods in a river system," *Biological Conservation*, vol. 197, pp. 131–138, 2016.

[5] M. C. Schmelzle and A. P. Kinziger, "Using occupancy modelling to compare environmental dna to traditional field methods for regional-scale monitoring of an endangered aquatic species," *Molecular ecology resources*, vol. 16, no. 4, pp. 895–908, 2016.

[6] Q. D. Wheeler, P. H. Raven, and E. O. Wilson, "Taxonomy: impediment or expedient?," 2004.

[7] P. Taberlet, E. Coissac, F. Pompanon, C. Brochmann, and E. Willerslev, "Towards next-generation biodiversity assessment using dna metabarcoding," *Molecular ecology*, vol. 21, no. 8, pp. 2045–2050, 2012.

[8] V. Elbrecht and F. Leese, "Can dna-based ecosystem assessments quantify species abundance? testing primer bias and biomass—sequence relationships with an innovative metabarcoding protocol," *PloS one*, vol. 10, no. 7, p. e0130324, 2015.

[9] E. Coissac, T. Riaz, and N. Puillandre, "Bioinformatic challenges for dna metabarcoding of plants and animals," *Molecular ecology*, vol. 21, no. 8, pp. 1834–1847, 2012.

[10] P. D. Hebert and T. R. Gregory, "The promise of dna barcoding for taxonomy," *Systematic biology*, vol. 54, no. 5, pp. 852–859, 2005.

[11] S. Ratnasingham and P. D. Hebert, "Bold: The barcode of life data system (http://www. barcodinglife. org)," *Molecular ecology notes*, vol. 7, no. 3, pp. 355–364, 2007.

[12] A. C. Raclariu, M. Heinrich, M. C. Ichim, and H. Boer, "Benefits and limitations of dna barcoding and metabarcoding in herbal product authentication," *Phytochemical Analysis*.

[13] Y. Ji, L. Ashton, S. M. Pedley, D. P. Edwards, Y. Tang, A. Nakamura, R. Kitching, P. M. Dolman, P. Woodcock, F. A. Edwards, *et al.*, "Reliable, verifiable and efficient monitoring of biodiversity via metabarcoding," *Ecology letters*, vol. 16, no. 10, pp. 1245–1257, 2013.

[14] P. Taberlet, E. Coissac, M. Hajibabaei, and L. H. Rieseberg, "Environmental dna," *Molecular ecology*, vol. 21, no. 8, pp. 1789–1793, 2012.

[15] M. W. Pedersen, S. Overballe-Petersen, L. Ermini, C. Der Sarkissian, J. Haile, M. Hellstrom, J. Spens, P. F. Thomsen, K. Bohmann, E. Cappellini, *et al.*, "Ancient and modern environmental dna," *Phil. Trans. R. Soc. B*, vol. 370, no. 1660, p. 20130383, 2015.

[16] A. Dell'Anno and C. Corinaldesi, "Degradation and turnover of extracellular dna in marine sediments: ecological and methodological considerations," *Applied and environmental microbiology*, vol. 70, no. 7, pp. 4384–4386, 2004.

[17] H. C. Rees, B. C. Maddison, D. J. Middleditch, J. R. Patmore, and K. C. Gough, "The detection of aquatic animal species using environmental dna–a review of edna as a survey tool in ecology," *Journal of Applied Ecology*, vol. 51, no. 5, pp. 1450–1459, 2014.

[18] G. Pietramellara, J. Ascher, F. Borgogni, M. Ceccherini, G. Guerri, and P. Nannipieri, "Extracellular dna in soil and sediment: fate and ecological relevance," *Biology and Fertility of Soils*, vol. 45, no. 3, pp. 219–235, 2009.

[19] A. Dell'Anno, B. Stefano, and R. Danovaro, "Quantification, base composition, and fate of extracellular dna in marine sediments," *Limnology and Oceanography*, vol. 47, no. 3, pp. 899–905, 2002.

[20] M. Hajibabaei, "The golden age of dna metasystematics," *Trends in genetics*, vol. 28, no. 11, pp. 535–537, 2012.

[21] A. Oulas, C. Pavloudi, P. Polymenakou, G. A. Pavlopoulos, N. Papanikolaou, G. Kotoulas, C. Arvanitidis, and l. Iliopoulos, "Metagenomics: tools and insights for analyzing next-generation sequencing data derived from biodiversity studies," *Bioinformatics and biology insights*, vol. 9, pp. BBI–S12462, 2015.

[22] M. Schirmer, U. Z. Ijaz, R. D'Amore, N. Hall, W. T. Sloan, and C. Quince, "Insight into biases and sequencing errors for amplicon sequencing with the illumina miseq platform," *Nucleic acids research*, vol. 43, no. 6, pp. e37–e37, 2015.

[23] K. Bohmann, A. Evans, M. T. P. Gilbert, G. R. Carvalho, S. Creer, M. Knapp, W. Y. Douglas, and M. De Bruyn, "Environmental dna for wildlife biology and biodiversity monitoring," *Trends in Ecology & Evolution*, vol. 29, no. 6, pp. 358–367, 2014.

[24] J. Herder, A. Valentini, E. Bellemain, T. Dejean, J. Delft, P. Thomsen, and P. Taberlet, "Environmental dna - a review of the possible applications for the detection of (invasive) species.," 06 2014.

[25] A. M. Polanowski, J. Robbins, D. Chandler, and S. N. Jarman, "Epigenetic estimation of age in humpback whales," *Molecular ecology resources*, vol. 14, no. 5, pp. 976–987, 2014.

[26] L. J. Clarke, J. M. Beard, K. M. Swadling, and B. E. Deagle, "Effect of marker choice and thermal cycling protocol on zooplankton dna metabarcoding studies," *Ecology and evolution*, vol. 7, no. 3, pp. 873–883, 2017.

[27] M. Kim and J. Chun, "16s rrna gene-based identification of bacteria and archaea using the eztaxon server," *Methods in Microbiology*, vol. 41, pp. 61–74, 2014.

[28] S. Creer, K. Deiner, S. Frey, D. Porazinska, P. Taberlet, W. K. Thomas, C. Potter, and H. M. Bik, "The ecologist's field guide to sequence-based identification of biodiversity," *Methods in Ecology and Evolution*, vol. 7, no. 9, pp. 1008–1018, 2016.

[29] M. L. Zepeda Mendoza, T. Sicheritz-Pontén, and M. T. P. Gilbert, "Environmental genes and genomes: understanding the differences and challenges in the approaches and software for their analyses," *Briefings in bioinformatics*, vol. 16, no. 5, pp. 745–758, 2015.

[30] L. J. Clarke, J. Soubrier, L. S. Weyrich, and A. Cooper, "Environmental metabarcodes for insects: in silico pcr reveals potential for taxonomic bias," *Molecular Ecology Resources*, vol. 14, no. 6, pp. 1160–1170, 2014.

[31] H. Vestheim and S. N. Jarman, "Blocking primers to enhance pcr amplification of rare sequences in mixed samples–a case study on prey dna in antarctic krill stomachs," *Frontiers in zoology*, vol. 5, no. 1, p. 12, 2008.

[32] O. S. Wangensteen and X. Turon, "Metabarcoding techniques for assessing biodiversity of marine animal forests," in *Marine Animal Forests*, pp. 445–473, Springer, 2017.

[33] M. Mouton, F. Postma, J. Wilsenach, and A. Botha, "Diversity and characterization of culturable fungi from marine sediment collected from st. helena bay, south africa," *Microbial ecology*, vol. 64, no. 2, pp. 311–319, 2012.

[34] M. Bálint, P.-A. Schmidt, R. Sharma, M. Thines, and I. Schmitt, "An illumina metabarcoding pipeline for fungi," *Ecology and evolution*, vol. 4, no. 13, pp. 2642–2653, 2014.

[35] M. Leray, N. Agudelo, S. C. Mills, and C. P. Meyer, "Effectiveness of annealing blocking primers versus restriction enzymes for characterization of generalist diets: unexpected prey revealed in the gut contents of two coral reef fish species," *PloS one*, vol. 8, no. 4, p. e58076, 2013.

[36] S. Boessenkool, L. S. Epp, J. Haile, E. Bellemain, M. Edwards, E. Coissac, E. Willerslev, and C. Brochmann, "Blocking human contaminant dna during pcr allows amplification of rare mammal species from sedimentary ancient dna," *Molecular ecology*, vol. 21, no. 8, pp. 1806–1815, 2012.

[37] A. Kumar and N. Chordia, "In silico pcr primer designing and validation," *PCR Primer Design*, pp. 143–151, 2015.

[38] K. Clark, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and E. W. Sayers, "Genbank," *Nucleic acids research*, vol. 44, no. D1, pp. D67–D72, 2015.

[39] J. Kans, "Entrez Programming Utilities Help ." https://www.ncbi.nlm.nih.gov/books/NBK179288/.

[40] R. C. Edgar, "Search and clustering orders of magnitude faster than blast," *Bioinformatics*, vol. 26, no. 19, pp. 2460–2461, 2010.

[41] K. Katoh, K. Misawa, K.-i. Kuma, and T. Miyata, "Mafft: a novel method for rapid multiple sequence alignment based on fast fourier transform," *Nucleic acids research*, vol. 30, no. 14, pp. 3059–3066, 2002.

[42] H. Yoon and T. Leitner, "Primerdesign-m: a multiple-alignment based multiple-primer design tool for walking across variable genomes," *Bioinformatics*, vol. 31, no. 9, pp. 1472–1474, 2014.

[43] J. Brodin, M. Krishnamoorthy, G. Athreya, W. Fischer, P. Hraber, C. Gleasner, L. Green, B. Korber, and T. Leitner, "A multiple-alignment based primer design algorithm for genetically highly variable dna targets," *BMC bioinformatics*, vol. 14, no. 1, p. 255, 2013.

[44] R. Owczarzy, A. V. Tataurov, Y. Wu, J. A. Manthey, K. A. McQuisten, H. G. Almabrazi, K. F. Pedersen, Y. Lin, J. Garretson, N. O. McEntaggart, *et al.*, "Idt scitools: a suite for analysis and design of nucleic acid oligomers," *Nucleic acids research*, vol. 36, no. suppl_2, pp. W163–W169, 2008.

[45] A. Larsson, "Aliview: a fast and lightweight alignment viewer and editor for large datasets," *Bioinformatics*, vol. 30, no. 22, pp. 3276–3278, 2014.

[46] A. Klindworth, E. Pruesse, T. Schweer, J. Peplies, C. Quast, M. Horn, and F. O. Glöckner, "Evaluation of general 16s ribosomal rna gene pcr primers for classical and next-generation sequencing-based diversity studies," *Nucleic acids research*, vol. 41, no. 1, pp. e1–e1, 2013.

[47] C. Quast, E. Pruesse, P. Yilmaz, J. Gerken, T. Schweer, P. Yarza, J. Peplies, and F. O. Glöckner, "The silva ribosomal rna gene database project: improved data processing and web-based tools," *Nucleic acids research*, vol. 41, no. D1, pp. D590–D596, 2012.

[48] A. Valentini, P. Taberlet, C. Miaud, R. Civade, J. Herder, P. F. Thomsen, E. Bellemain, A. Besnard, E. Coissac, F. Boyer, *et al.*, "Next-generation monitoring of aquatic biodiversity using environmental dna metabarcoding," *Molecular Ecology*, vol. 25, no. 4, pp. 929–942, 2016.

[49] "X11 - X windowing system ." http://toastytech.com/guis/remotex11.html.

[50] E. Robert, "Defining and interpreting OTUs." https://www.drive5.com/usearch/manual/otus.html.

[51] N.-P. Nguyen, T. Warnow, M. Pop, and B. White, "A perspective on 16s rrna operational taxonomic unit clustering using sequence similarity," *NPJ biofilms and microbiomes*, vol. 2, p. 16004, 2016.

[52] F. Mahé, T. Rognes, C. Quince, C. de Vargas, and M. Dunthorn, "Swarm: robust and fast clustering method for amplicon-based studies," *PeerJ*, vol. 2, p. e593, 2014.

[53] O. Holovachov, Q. Haenel, S. J. Bourlat, and U. Jondelius, "Taxonomy assignment approach determines the efficiency of identification of otus in marine nematodes," *Royal Society open science*, vol. 4, no. 8, p. 170315, 2017.

[54] K. Gdanetz, G. M. N. Benucci, N. V. Pol, and G. Bonito, "Constax: A tool for improved taxonomic resolution of environmental fungal its sequences," *BMC bioinformatics*, vol. 18, no. 1, p. 538, 2017.

[55] V. Vasselon, A. Bouchez, F. Rimet, S. Jacquet, R. Trobajo, M. Corniquel, K. Tapolczai, and I. Domaizon, "Avoiding quantification bias in metabarcoding: application of a cell biovolume correction factor in diatom molecular biomonitoring," *Methods in Ecology and Evolution*, vol. 9, no. 4, pp. 1060–1069, 2018.

[56] E. Britannica, "Defining and interpreting OTUs." https://www.britannica.com/science/biogeographic-region/Components-of-species-diversity-species-richness-and-relative-abundance#ref588341.

[57] N. T. Evans, B. P. Olds, M. A. Renshaw, C. R. Turner, Y. Li, C. L. Jerde, A. R. Mahon, M. E. Pfrender, G. A. Lamberti, and D. M. Lodge, "Quantification of mesocosm fish

and amphibian species diversity via environmental dna metabarcoding," *Molecular ecology resources*, vol. 16, no. 1, pp. 29–41, 2016.

[58] H. M. Bik, D. L. Porazinska, S. Creer, J. G. Caporaso, R. Knight, and W. K. Thomas, "Sequencing our way towards understanding global eukaryotic biodiversity," *Trends in ecology & evolution*, vol. 27, no. 4, pp. 233–243, 2012.

[59] S. Andrews, "FASTQC." http://www.bioinformatics.babraham.ac.uk/projects/fastqc/.

[60] E. Bioinformatics, "Why does the per base sequence quality decrease over the read in Illumina?." https://www.ecseq.com/support/ngs/why-does-the-sequence-quality-decrease-over-the-read-in-illumina.

[61] A. M. Bolger, M. Lohse, and B. Usadel, "Trimmomatic: a flexible trimmer for illumina sequence data," *Bioinformatics*, vol. 30, no. 15, pp. 2114–2120, 2014.

[62] A. Bankevich, S. Nurk, D. Antipov, A. A. Gurevich, M. Dvorkin, A. S. Kulikov, V. M. Lesin, S. I. Nikolenko, S. Pham, A. D. Prjibelski, *et al.*, "Spades: a new genome assembly algorithm and its applications to single-cell sequencing," *Journal of computational biology*, vol. 19, no. 5, pp. 455–477, 2012.

[63] S. I. Nikolenko, A. I. Korobeynikov, and M. A. Alekseyev, "Bayeshammer: Bayesian clustering for error correction in single-cell sequencing," in *BMC genomics*, vol. 14, p. S7, BioMed Central, 2013.

[64] A. P. Masella, A. K. Bartram, J. M. Truszkowski, D. G. Brown, and J. D. Neufeld, "Pandaseq: paired-end assembler for illumina sequences," *BMC bioinformatics*, vol. 13, no. 1, p. 31, 2012.

[65] F. Boyer, C. Mercier, A. Bonin, Y. Le Bras, P. Taberlet, and E. Coissac, "obitools: a unix-inspired software package for dna metabarcoding," *Molecular ecology resources*, vol. 16, no. 1, pp. 176–182, 2016.

[66] T. Rognes, T. Flouri, B. Nichols, C. Quince, and F. Mahé, "Vsearch: a versatile open source tool for metagenomics," *PeerJ*, vol. 4, p. e2584, 2016.

[67] R. C. Edgar, "Uparse: highly accurate otu sequences from microbial amplicon reads," *Nature methods*, vol. 10, no. 10, p. 996, 2013.

[68] F. Mahé, T. Rognes, C. Quince, C. de Vargas, and M. Dunthorn, "Swarm v2: highly-scalable and high-resolution amplicon clustering," *PeerJ*, vol. 3, p. e1420, 2015.

[69] E. Robert, "Uchime in practice."

[70] X. Hao, R. Jiang, and T. Chen, "Clustering 16s rrna for otu prediction: a method of unsupervised bayesian clustering," *Bioinformatics*, vol. 27, no. 5, pp. 611–618, 2011.

[71] A. Lanzén, S. L. Jørgensen, D. H. Huson, M. Gorfer, S. H. Grindhaug, I. Jonassen, L. Øvreås, and T. Urich, "Crest–classification resources for environmental sequence tags," *PloS one*, vol. 7, no. 11, p. e49334, 2012.

[72] A. Lanzen, "Crest - classification resources for environmnetal sequence tags, is a collection of software and databases for taxonomic classification of environmental marker genes from sequencing-based community profiling studies.."

[73] "Silva database release_132."

[74] T. Z. DeSantis, P. Hugenholtz, N. Larsen, M. Rojas, E. L. Brodie, K. Keller, T. Huber, D. Dalevi, P. Hu, and G. L. Andersen, "Greengenes, a chimera-checked 16s rrna gene database and workbench compatible with arb," *Applied and environmental microbiology*, vol. 72, no. 7, pp. 5069–5072, 2006.

[75] U. Kõljalg, K.-H. Larsson, K. Abarenkov, R. H. Nilsson, I. J. Alexander, U. Eberhardt, S. Erland, K. Høiland, R. Kjøller, E. Larsson, *et al.*, "Unite: a database providing web-based methods for the molecular identification of ectomycorrhizal fungi," *New Phytologist*, vol. 166, no. 3, pp. 1063–1068, 2005.

[76] R. J. E. Alves, B. Q. Minh, T. Urich, A. Haeseler, and C. Schleper, "Unifying the global phylogeny and environmental distribution of ammonia-oxidising archaea based on amoa genes," *Nature communications*, vol. 9, no. 1, p. 1517, 2018.

[77] L. Czech and A. Stamatakis, "Scalable methods for post-processing, visualizing, and analyzing phylogenetic placements," *bioRxiv*, p. 346353, 2018.

[78] A. Kozlov, "amkozlov/raxml-ng: Raxml-ng v0.6.0 beta," June 2018.

[79] P. Barbera, A. M. Kozlov, L. Czech, B. Morel, D. Darriba, T. Flouri, and A. Stamatakis, "Epa-ng: Massively parallel evolutionary placement of genetic sequences," *bioRxiv*, p. 291658, 2018.

[80] S. A. Berger and A. Stamatakis, "Papara 2.0: a vectorized algorithm for probabilistic phylogeny-aware alignment extension," *Heidelberg Institute for Theoretical Studies, http://sco. h-its. org/exelixis/publica tions. html. Exelixis-RRDR-2012-2015*, 2012.

[81] I. Letunic and P. Bork, "Interactive tree of life (itol): an online tool for phylogenetic tree display and annotation," *Bioinformatics*, vol. 23, no. 1, pp. 127–128, 2006.

[82] Q. Wang, G. M. Garrity, J. M. Tiedje, and J. R. Cole, "Naive bayesian classifier for rapid assignment of rrna sequences into the new bacterial taxonomy," *Applied and environmental microbiology*, vol. 73, no. 16, pp. 5261–5267, 2007.

[83] R. J. Machida, M. Leray, S.-L. Ho, and N. Knowlton, "Metazoan mitochondrial gene sequence reference datasets for taxonomic assignment of environmental samples," *Scientific data*, vol. 4, p. 170027, 2017.

[84] I. Lagkouvardos, S. Fischer, N. Kumar, and T. Clavel, "Rhea: a transparent and modular r pipeline for microbial profiling based on 16s rrna gene amplicons," *PeerJ*, vol. 5, p. e2836, 2017.

[85] C. Pavloudi, J. B. Kristoffersen, A. Oulas, M. De Troch, and C. Arvanitidis, "Sediment microbial taxonomic and functional diversity in a natural salinity gradient challenge remane's "species minimum" concept," *PeerJ*, vol. 5, p. e3687, 2017.

[86] I. Bista, G. R. Carvalho, K. Walsh, M. Seymour, M. Hajibabaei, D. Lallias, M. Christmas, and S. Creer, "Annual time-series analysis of aqueous edna reveals ecologically relevant dynamics of lake ecosystem biodiversity," *Nature communications*, vol. 8, p. 14087, 2017.

[87] "Ena file downloader version 1.2."

[88] P. Cingolani, R. Sladek, and M. Blanchette, "Bigdatascript: a scripting language for data pipelines," *Bioinformatics*, vol. 31, no. 1, pp. 10–16, 2014.

[89] Wangensteen, "Defining and interpreting OTUs." https://www.physalia-courses.org/news/n11/.

[90] K. Deiner, H. M. Bik, E. Mächler, M. Seymour, A. Lacoursière-Roussel, F. Altermatt, S. Creer, I. Bista, D. M. Lodge, N. de Vere, *et al.*, "Environmental dna metabarcoding: transforming how we survey animal and plant communities," *Molecular ecology*, vol. 26, no. 21, pp. 5872–5895, 2017.

[91] J. F. Swift, R. F. Lance, X. Guan, E. R. Britzke, D. L. Lindsay, and C. E. Edwards, "Multifaceted dna metabarcoding: Validation of a noninvasive, next-generation approach to studying bat populations," *Evolutionary Applications*, 2018.

[92] C. Vacher, A. Tamaddoni-Nezhad, S. Kamenova, N. Peyrard, Y. Moalic, R. Sabbadin, L. Schwaller, J. Chiquet, M. A. Smith, J. Vallance, *et al.*, "Learning ecological networks from next-generation sequencing data," in *Advances in Ecological Research*, vol. 54, pp. 1–39, Elsevier, 2016.

[93] C. J. Macgregor, J. J. Kitson, R. Fox, C. Hahn, D. H. Lunt, M. J. Pocock, and D. M. Evans, "Construction, validation and application of nocturnal pollen transport networks in an agro-ecosystem: a comparison using microscopy and dna metabarcoding," *bioRxiv*, p. 325084, 2018.

[94] D. M. Evans, J. J. Kitson, D. H. Lunt, N. A. Straw, and M. J. Pocock, "Merging dna metabarcoding and ecological network analysis to understand and build resilient terrestrial ecosystems," *Functional ecology*, vol. 30, no. 12, pp. 1904–1916, 2016.

Part V

Appendix

1 Blocking primer design *in silico*

Important commands used in the designing of blocking primers are following, as those performed to acquire the fungal sequences of 16S gene from GenBank.

```
esearch -db nuccore -query "16S [ALL] fungi [FILT]" | efetch -format fasta
```

In the command above the function “esearch” performs an Entrez search using terms in indexed fields, in order to explore the database “nuccore” for all the entries that include “16S” and that are related to Fungi ("fungi" in this query is a filter that we ask to be positive in an entry, in order to be returned as a result in our search). Finally, we use the function “efetch” that downloads records or reports in a designated format and we choose the format fasta for our sequences.

Something worth-mentioning is the fact that when this command was executed for the first time, it returned 7632 sequences while only some months later, would give a significantly greater number of sequences (7910).

In order to remove sequences that could not contribute in our research and that could be noticed by their title, commands as the following were executed: `all_head.txt |`

```
grep -vi internal | grep -vi ITS | grep -vi intergenic | grep -vi spacer
> no_internal_no_its.txt
```

For finding the OTUs for the species from which there were more than 5 sequences, USEARCH was used:

```
./usearch7.0.1090_i86linux32 -cluster_otus /home/haris/fungi_seqs/species/
each_species_seperately/uniques.fasta -otus /home/haris/siga.fasta
```

Finally, in order to get any information possible from the meta - data of the sequences that the .gb file carried, we ran a series of "awk" commands like:

```
awk 'BEGIN RS="//" !/(soil|Soil|SOIL)/ print ">"$0' fetch_to_check_for_soil.gb
```

In this way we managed to reduce in a significant level the number of sequences that finally were used in for the alignment and the blocking primer design.

In the case of COI gene, the extra filtering step was for removing the long sequences was achieved through the command:

```
awk 'BEGIN RS=">" length($0)<2500 print ">"$0' COI_seqs_for_alignment.fa
> COI_seqs_for_alignment_no_big.fa
```

At the evaluation step we had to find those sequences that had been matched with both the forward and reversed primer. In order to achieve that, the next command was executed:

```
comm -12 f1_0_all_sorted.txt r1_0_all_sorted.txt
```

//

2 P.E.M.A: a Pipeline for EnvironmentalDNA Metabarcoding Analysis

Command ran in RAxML-ng for building the tree needed for Rhea:
raxmlHPC -f a -p 12345 -s aligned_otus.fasta -x 54321 -# 100 -m GTRGAMMA

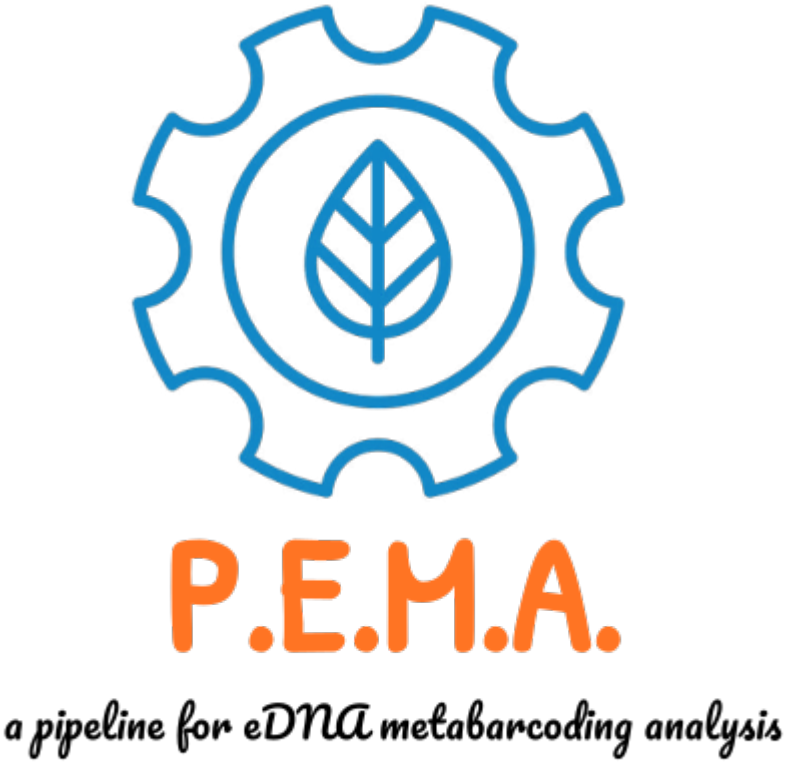
sample	**number of sequences that were assigned to species level**	**number of sequences after pre-processing**
ERR1308201	131	15171
ERR1308202	5	8781
ERR1308203	0	3458
ERR1308204	0	15324
ERR1308205	2	9477
ERR1308206	41	13036
ERR1308207	59	9945
ERR1308208	37	13049
ERR1308209	24	14948
ERR1308210	0	29
ERR1308211	0	37
ERR1308233	39	12486
ERR1308234	16	10418
ERR1308235	83	16477
ERR1308236	27	14999
ERR1308237	67	13429
ERR1308238	339	25635
ERR1308239	32	17606
ERR1308240	0	20

ERR1308241	2	15
ERR1308242	0	6487
ERR1308243	0	26
ERR1308244	0	8
ERR1308245	0	18
ERR1308246	555	8832
ERR1308247	58	9755
ERR1308248	57	7685
ERR1308249	712	10627
ERR1308250	4954	7885
ERR1308251	140	19617
ERR1308252	2	19640
ERR1308253	3946	25970
ERR1308287	0	34288
ERR1308288	0	44036
ERR1308289	0	30451
ERR1308290	2	51755
ERR1308291	0	69353
ERR1308292	0	55756
ERR1308293	0	53520
ERR1308294	0	43553
ERR1308295	0	28442
ERR1308296	2	37814
ERR1308297	0	20105
ERR1308298	0	48287
ERR1308299	0	15958
ERR1308300	0	22884
ERR1308301	0	16535
ERR1308302	0	35302
ERR1308303	26	95848
ERR1308304	0	1482
ERR1308305	0	748
ERR1308306	0	682
ERR1308307	0	39
ERR1308308	0	488
ERR1308309	20	32382
ERR1308310	0	70
ERR1308311	0	1090

ERR1308312	10	47079
ERR1308313	8	23637
ERR1308314	2379	47807
ERR1308315	28	30356
ERR1308316	0	1310
ERR1308317	0	3338
ERR1308318	0	4664

sample	**number of sequences that were assigned to any taxon level with >= 97%**	**number of sequences after pre-processing**
ERR1308201	6821	15171
ERR1308202	5407	8781
ERR1308203	503	3458
ERR1308204	8046	15324
ERR1308205	6371	9477
ERR1308206	10863	13036
ERR1308207	650	9945
ERR1308208	1789	13049
ERR1308209	18673	14948
ERR1308210	0	29
ERR1308211	15	37
ERR1308233	1700	12486
ERR1308234	9344	10418
ERR1308235	10375	16477
ERR1308236	5617	14999
ERR1308237	2625	13429
ERR1308238	31997	25635
ERR1308239	8835	17606
ERR1308240	0	20
ERR1308241	2	15
ERR1308242	8223	6487
ERR1308243	0	26
ERR1308244	0	8
ERR1308245	0	18
ERR1308246	2859	8832
ERR1308247	12594	9755

ERR1308248	1127	7685
ERR1308249	7251	10627
ERR1308250	5333	7885
ERR1308251	9266	19617
ERR1308252	13804	19640
ERR1308253	11020	25970
ERR1308287	2238	34288
ERR1308288	6204	44036
ERR1308289	796	30451
ERR1308290	1645	51755
ERR1308291	4261	69353
ERR1308292	401	55756
ERR1308293	1053	53520
ERR1308294	270	43553
ERR1308295	327	28442
ERR1308296	28278	37814
ERR1308297	770	20105
ERR1308298	10259	48287
ERR1308299	394	15958
ERR1308300	63	22884
ERR1308301	106	16535
ERR1308302	1552	35302
ERR1308303	15028	95848
ERR1308304	0	1482
ERR1308305	2	748
ERR1308306	4	682
ERR1308307	0	39
ERR1308308	0	488
ERR1308309	66	32382
ERR1308310	0	70
ERR1308311	27	1090
ERR1308312	1689	47079
ERR1308313	361	23637
ERR1308314	2807	47807
ERR1308315	242	30356
ERR1308316	4	1310
ERR1308317	0	3338
ERR1308318	4	4664



P.E.M.A. : a Pipeline for Environmental DNA Metabarcoding Analysis for the 16S and COI marker genes

P.E.M.A. is a pipeline for two marker genes, **16S rRNA** (microbes) and **COI** (Eukaryotes). As input, P.E.M.A. accepts .fastq files as returned by Illumina sequencing platforms. P.E.M.A. processes the reads from each sample and **returns an OTU-table with the taxonomies** of the taxa found and their abundances in each sample. It also returns statistics and a FASTQC diagram about the quality of the reads for each sample. Finally, in the case of 16S, P.E.M.A. returns **alpha and beta diversities**, and make correlations between samples. The last step is facilitated by Rhea, a set of R scripts for downstream 16S amplicon analysis of microbial profiles.

In the COI case, two clustering algorithms can be performed by P.E.M.A. (CROP and SWARM), while in the 16S, two approaches for taxonomy assignment are supported: alignment- and phylogenetic-based. For the latter, a reference tree with 1000 taxa was created using SILVA_132_SSURef, EPA-ng and RaxML-ng.

Getting Started

P.E.M.A. is able to run either on a HPC environment (server, cluster etc) or on a simple PC of your own. However, we definitely suggest to run it on an HPC environment. A powerful server or a cluster, even better, is necessary, as P.E.M.A. would take ages in a common PC.

There is one **major difference** between running P.E.M.A. on your own PC than running it on a HPC environment. In the first case, P.E.M.A. runs through **Docker**, while in the latter one, it runs through **Singularity**.

On the next chapters, you can find how to install P.E.M.A. in each case as well as an example of running it.

Running P.E.M.A. is exactly **the same** procedure in both oh these cases.

P.E.M.A on HPC

P.E.M.A. is best to run on HPC (server, cluster, cloud). Usually Environmental data are quite large and the whole process has huge computational demands. To get P.E.M.A. running on your HPC you need just to do the followings.

Prerequisites

Singularity is a free, cross-platform and open-source computer program that performs operating-system-level virtualization also known as containerization. One of the main uses of Singularity is to bring containers and reproducibility to scientific computing and the high-performance computing (HPC) world

Singularity, needs a Linux system to run .

Installing

After you install Singularity in your environment and open it, you need to download P.E.M.A.'s image from Docker Hub, by running the command:

```
singularity pull docker://hariszaf/pema
```

Now you have P.E.M.A. on your environment and the only thing that is left to do, is to fulfill the **parapeters.tsv** file (see below, on "Parameters' file" section) and run P.E.M.A.

Running P.E.M.A.

Singularity allows to use a job scheduler that allocates compute resources on clusters and at the same time, works as a queuing system, as **Slurm**. This way you are able to create a job as you useally do in your system and after setting the parameters' file as you want to, run P.E.M.A. as a job on your cluster.

Example

```
#SBATCH --partition=batch
#SBATCH --nodes=2
#SBATCH --ntasks-per-node=20
```

```
#SBATCH --mem=
# Memory per node specification is in MB. It is optional.
# The default limit is 3000MB per core.
#SBATCH --job-name="testPema"
#SBATCH --output=PEMA.output
#SBATCH --mail-user=haris-zafr@hcmr.gr
#SBATCH --mail-type=ALL
#SBATCH --requeue

singularity exec ~/ubuntu.img echo "Hey, I'm running ubuntu"
```

In the above job, we set HCMR's cluster "Zorba", to run P.E.M.A. in 2 nodes, with 20 cores in each of those.

P.E.M.A on a simple PC

Prerequisites

To run P.E.M.A. in a simple PC on your own environment, you first need to install Docker (https://docs.docker.com/install/), in case you do not already have it.

You should check your software version. Docker is avalable for all Windows, Mac and Linux. However, in case of Windows and Mac, you might need to install Docker toolbox instead (https://docs.docker.com/toolbox/), if your System Requirements are not the ones mentioned below.

System Requirements

```
**__Windows 10 64bit__**:
Pro, Enterprise or Education (1607 Anniversary Update, Build 14393 or later).
Virtualization is enabled in BIOS. Typically, virtualization is enabled by default.
This is different from having Hyper-V enabled. For more detail see Virtualization
must be enabled in Troubleshooting.
CPU SLAT-capable feature.
At least 4GB of RAM.

**__Mac__**
Mac hardware must be a 2010 or newer model, with Intel’s hardware support for
memory management unit (MMU)
virtualization, including Extended Page Tables (EPT) and Unrestricted Mode. You can
check to see if your machine
has this support by running the following command in a terminal:
sysctl kern.hv_support macOS El Capitan 10.11 and newer macOS releases are
supported.
We recommend upgrading to the latest version of macOS.
At least 4GB of RAM
VirtualBox prior to version 4.3.30 must NOT be installed (it is incompatible with
Docker for Mac).
If you have a newer version of VirtualBox installed, it’s fine.
```

Installing

After you install Docker in your environment and open it, the only thing you need to do, is to download P.E.M.A.'s image, by running the command:

```
docker pull hariszaf/pema
```

P.E.M.A. is a quite large image (~2Gb) so it will take a while until it is downloaded in your computer system.

Running P.E.M.A.

Running P.E.M.A. has two discrete steps.

Step 1 - Build a Docker container

At first, you need to let Docker have access in your dataset. For this you need to run this command, specifying the path to where your data is stored, i.e. changing the path_to_my_data accordingly:

```
docker run -it vol -v /<path_to_my_data>:/vol_myData pema
```

After you run the command above, you have now built a Docker container, in which you can work with P.E.M.A.

P.E.M.A. gives you the opportunity, among others, to BLAST your data, in case you want to.

In this and only in this case, you need to tell P.E.M.A. where to find your BLAST database. So, in this case, you skip the previous command, and execute the commands below,specifying the path to where your data and the BLAST database are stored, i.e. changing the path_to_my_data and the path_to_BLAST_Database accordingly:

```
docker run -it --name vol -v /<path_to_my_data>:/vol_myData -v
/<path_to_BLAST_Database>:/vol_myDataBase PEMA_image

docker run -it --rm --name foo --volumes-from=vol
```

Step 2 - Run P.E.M.A.

To run P.E.M.A. you first need to set all parameters the way they should be, depending on your dataset and your experiment.

To do so, run the command below and set the parameters in it:

```
nano parameters.csv
```

For more details about the #parameters.csv#, please check on the manual for the parapeter's file.

Finally, when all the above are set, the only thing remaining to do, is to run P.E.M.A.

```
./PEMA_docker_version.bds
```

P.E.M.A. is now running and it depends on your computer (or server or cluster), on the size of your data, as well as on the parameters you chose, how long it will take.

When

In order to get the output file in your computer, you just need to copy it from the Docker container you are working on. To do so, you just need to see the **id** of your container, simply by typing:

```
docker ps -a
```

and then, copying anything you want to, from a single file to the whole folder, with a command like this:

```
docker cp <contaier_ID>:/path/to/what/you/need/to/copy/
/path/to/the/directory/you/want/to/copy/it/to/
```

Please, keep in mind that when you want to copy a whole directory, then you always have to put "/" in the end of the path that describes where the folder is located.

Parameters' file

The most crucial component in running P.E.M.A. is the parameters' file. This is located in the same directory as P.E.M.A. does and the user needs to fill it **every time** P.E.M.A. is about to be called.

So, here is the **parameters.tsv** file as it looks like, in a study case of our own. The user has to set it the way it fits to his own data.

Acknowledgments

P.E.M.A. uses a series of tools, datasets as well as Big Data Script language. We have to thank all of these groups. The tools & databases that PEMA uses are : * FASTQC - https://www.bioinformatics.babraham.ac.uk/projects/fastqc/ * Trimmomatic - http://www.usadellab.org/cms/?page=trimmomatic * SPAdes - http://cab.spbu.ru/software/spades/ * BayesHammer - included in SPAdes * OBITools - https://pythonhosted.org/OBITools/welcome.html * USEARCH - https://www.drive5.com/usearch/ * BLAST- https://blast.ncbi.nlm.nih.gov/Blast.cgiCMD=Web&PAGE_TYPE=BlastDocs&DOC_TYPE=Download *

CREST - https://github.com/lanzen/CREST * SILVA db - https://www.arb-silva.de/no_cache/download/archive/current/Exports/ * RAxML - https://sco.h-its.org/exelixis/web/software/raxml/index.html * RAxML -ng - https://github.com/amkozlov/raxml-ng * EPA-ng - https://github.com/Pbdas/epa-ng * SWARM - https://github.com/torognes/swarm * CROP - https://github.com/tingchenlab/CROP * VSEARCH-2.7.1 - https://github.com/torognes/vsearch/releases/tag/v2.7.1 * RDPClassifier - https://github.com/rdpstaff/classifier (RPDtools are required in order to execute RDPClassifier)

```
#################################################################################################
################################   P.E.M.A. 's PARAMETERS   ###################################
#################################################################################################
#
#
# In this file there are all the parameters that NEED TO BE ASSIGNED every time you need P.E.M.A. to run!
# That does not mean that these parameters that we have here, are the only  parameters of the tools P.E.M.A.
# uses! As you already may know, the combinations are infinite!
# Hence, we encourage you the most to study the manual of each tool and make them as good as possible for
# your SPECIFIC experiment.
#
# Every line either starts with a "#" or not. All lines that do not start with a "#" are variables that P.E.M.A. asks
# for. You have to set all these parameters according to your data and your experiment.
#
# We chose to have a set of parameters for some tools (mainly for Trimmomatic) as 'by default'.
# That is because in some cases, plenty of time is required in order to have a correct set of those.
# In the link next to each tool, you can find further information about its parameters.
#
#
# YOU NEED TO BE REALLY CAREFUL WHEN YOU FILL THIS FILE !!
#
#
# From each variable you have to leave EXACTLY ONE (1) TAB and then fill the parameter as you wish.
#
#
#################################################################################################
#################################################################################################
#
# This is where the parameters to set are:
#
# Just to test that the parameters are assigned right in our main script:
#  This is just a check that P.E.M.A. is able to read this file. Please continue.
#
sources             my_parameters_work_just_fine!
#
#
#
# Give in your each uniq experiment a NAME, so a single output file will be created for each of them.
# ATTENTION! If you do not change the name of the output file, and run a second analysis with the same
# name, if you have not move the  first one to another path, then it will be overwritten.
#
#
outputFile          put_here_the_name_of_your_output_file
#
#
#################################################################
#########################   blastqc   ###########################
#################################################################
#
#
## no parameters provided here at all!
#
#
#################################################################
```

```
##################### trimmomatic   ########################
#################################################################
# url: http://www.usadellab.org/cms/uploads/supplementary/Trimmomatic/TrimmomaticManual_V0.32.pdf
#
#
############## for MAXINFO ####################
#
# Performs an adaptive quality trim, balancing the benefits of retaining longer reads against the costs of
# retaining bases with errors. The values it takes, can be either 'Y'(Yes) or 'N'(No).
#
#
maxInfo            Y
#
#
# The parameter "targetLength" specifies the read length which is likely to allow the
# location of the read within the target sequence to be determined
#
#
targetLength       200
#
#
# This value, which should be set between 0 and 1, specifies the balance between preserving as much read '
# length as possible vs.  removal of incorrect bases. A low value of this parameter (<0.2) favours longer reads,
# while a high value (>0.8) favors read correctness.
#
#
strictness         0.3
#
#
############# for ILLUMINACLIP ##################
#
# Specifies the path to a fasta file containing all the adapters, PCR sequences etc.
# The naming of the various sequences within this file determines how they are used.
#
#
adapters    TruSeq2-PE.fa
#
#
# "seedMismatches" specifies the maximum mismatch count which will still allow a full match to be performed
#
#
seedMismatches    1
#
#
# Specifies how accurate the match between the two 'adapter ligated' reads must be for PE palindrome
# read alignment.
#
#
palindromeClipThreshold         30
#
#
## Specifies how accurate the match between any adapter etc. sequence must be against a read.
#
#
```

```
simpleClipThreshold    10
#
#
############ for LEADING  ##########################
#
# Cut bases off the start of a read, if below a threshold quality
# Remove low quality bases from the beginning. As long as a base has a value below this threshold the base
# is  removed and the next base will be investigated.
# Its quality: Specifies the minimum quality required to keep a base.
#
#
leading    3
#
#
############ for TRAILING  ############################
#
# Remove low quality bases from the end. As long as a base has a value below this
# from the 3‟ prime end would be base preceding the just removed base) will be investigated.
# Specifies the minimum quality required to keep a base.
#
#
trailing3
#
#
############  for MINLEN  ###################################
#
# This module removes reads that fall below the specified minimal length.
# The value it takes, specifies the minimum quality required to keep a base.
#
#
minlen 100
#
#
#
#####################################################################
#################  BayesHammer  ###################################
#####################################################################
#
## no parameters provided here at all!
#
#####################################################################
###################    PANDAseq    ##################################
#####################################################################
# url:  https://storage.googleapis.com/pandaseq/pandaseq.html
#
# PANDAseq has more than one merging algorithms. Here, we set the algorithm used for assembly
# pear --> uses the formula described in the PEAR paper (Zhang 2013), optionally with the probability
# of a random base (q) provide
# simple_bayesian --> uses the formula described in the original paper (Masella 2012), optionally with
# an error estimation (ε) provided.
# Other options are stich, flash and more that you can fing in the above link.
#
#
spadesAlgorithm        simple_bayesian
```

```
#
#
### In our command, we also have "-a" that strips the primers after assembly, rather than before.
### And "-B" that allows input sequences to lack a barcode/tag, as well.
#
#########################################################################
################  DEREPLICATE me ti voitheia ton OBITools !  ##########
#########################################################################
# url: https://pythonhosted.org/OBITools/
#
#
## no parameters here!
#
#
#########################################################################
#////////////////////////////////////////////////////////////////////////////////
############################## GENE -  dependent parameters ##########
#////////////////////////////////////////////////////////////////////////////////
#########################################################################
#
# The marker gene you have is really important for both the clustering & chimera removal procedure
# and the taxonomy assignment. By default, the pipeline runs for 16S. Substitute with 'COI' if COI is your
# marker gene.
# Write down your gene after character "_" without please erasing it! (e.g. "gene    gene_16S").
#
#
gene       gene_16S
#
#
# You might want to download the last version of SILVA database and handle it with the primers you have used
# in your experiment in order to get an even better assignment. You can find SILVA 132 already in PEMA/tools.
# Hence, by default getting_silva is FALSE. You have to turn it to TRUE in order to do so.
gettingSilva       FALSE
primerF            "
primerR            "
#
#
# If your marker gene is 16S, you can choose between 2 different approaches of taxonomy assignment (alignment
&
# phylogenetic based) by  default, you get an alignment based taxonomy assignment - set as 'alignment' -  which
# is based on SILVA and CREST.
# However you can also get a phylogenetic based assignment, by putting 'phylogeny' in this parameter.
#
#
taxonomyAssignmentMethod              alignment
#
#
# If your marker gene is  COI, you can choose between 2 different approaches of clustering.
# Depending on which of them you choose you get either a robust output in a short time (SWARM)
# or a non-robust output (CROP) that requires quite much more time. CROP is a bayesian
# algorithm and that is why its output in non-robust. By default, SWARM algorithm runs for the clustering.
#You have to change in to 'CROP' if you want the CROP algorithm to do the clustering step.
#
#
```

```
#
clusteringAlgo          algo_SWARM
#
#
# In case of SWARM, the user needs to specify the value of "d" parameter. "d" is the number of mismatches and
# it is the most important parameter of SWARM algorithm. By default "d" equals to 1 (best value for d, according
# to its authors).
# You are free to set it as you want but a large "d" will possibly give you not so trustful results.
#
#
d        1   1
#
#
#########################################################################
########################   BLAST   ##############################
#########################################################################
#
## Would you like to get one taxonomy assignment using normal BLAST? It can be either 'Yes' or 'No'.
#
#
blast    N
#
#
#
# Depending on the environment you are working on, you might need to set the path where your
# Blast Database is located.
#
#
blastDatabase    /mnt/big/blastdb/nt/nt/nt
#
#
#  If your marker gene is 16S and you wish to use Rhea in order to analyse your returned data
#  then you need to create a tree using these OTUs. If you do wish so, set 'preparingForRhea' as 'Yes'
#
#
forRhea             N
#
#
# Hence please give us the path where RAxML is located on your cluster
# If you need to make a tree for running Rhea, you will need RAxML.
#
#
raxmlPath       /usr/bin/raxmlHPC
#
#
# If you do not run P.E.M.A. on Docker and you have R elsewhere, please set the path another way.
# However, ONLY then, this parameter needs to change.
#
#
pathForR        /usr/bin/R
#
#
# In case you wish for Beta-diversity from Rhea, then you have to set a categorical variable.
# We should mention here, that P.E.M.A has already created the tree Rhea needs from its first step.
```

```
# Hence, the user now can use Rhea scripts, even out of P.E.M.A
# Hence, you have to set 'categoricalVariable' as 'Y' and then go to the Rhea script for Beta-diversity to set
# which variable you want as categorical as all needed files are made.
#
#
categoricalVariable     Y
#
#
# Give to PEMA the percentage of similarity over which you want to keep taxonomy as for sure taxa
#
#
percentageOfSimilarity            91
#
#
# Finally, do you want your raw data to be removed in another file and empty the "rawData" file and all the
# checkpoints of PEMA to be also in an extra folder.
# Swich 'Yes' to 'No' if you wish so.
# Be very careful when you do that, as you need to remember that if you want to analyze another dataset
# through P.E.M.A  you will have to remove the first one manually.
#
#
emptyRawDataFile       Y
emptyCheckpoints       Y
```