End-to-End Deep Learning and Subgroup discovery approaches to learn from metagenomics data

Maxence Queyrel

SORBONNE UNIVERSITÉ

Doctoral School: EDITE de Paris (ED130)
Speciality: Computer Science

Machine learning in bioinformatics

End-to-End Deep Learning and Subgroup discovery approaches to learn from metagenomics data

Maxence Queyrel

Reviewers Jean-Philippe Vert, University Professor, Google Brain, Mines Paris-Tech
Edoardo Pasolli, University Professor, Dipartimento di Agraria University di Napoli

Examiners Alessandra Carbone, University Professor, Sorbonne University
Blaise Hanczar, University Professor, IBISC, Paris-Saclay University

Supervisor Edi Prifti, Researcher Director, IRD UMMISCO, Sorbonne University

Directors Jean-Daniel Zucker, Researcher Director, IRD UMMISCO, Sorbonne University
Alexandre Templier, Chairman, Quinten

Co-director Karine Clément, University Professor, NUTRIOMICSS, Sorbonne University

October 13 2021

Maxence QUEYREL

End-to-End Deep Learning and Subgroup discovery approaches to learn from metagenomics data

Computer Science, Machine learning in bioinformatics

Reviewers: Jean-Philippe Vert and Edoardo Pasolli

Examinors: Alessandra Carbone and Blaise Hanczar

Supervisor: Edi Prifti

Director: Jean-Daniel Zucker and Alexandre Templier

Co-Director: Karine Clément

SORBONNE UNIVERSITÉ

Doctoral School: EDITE de Paris (ED130)

Research Institute: IRD / ICAN

Department/Laboratory: UMMISCO

4 Place Jussieu, 75005 Paris V

Date of defense: October 13 2021

Acknowledgment

This thesis took place within the framework of a CIFRE between the laboratory UMMISCO (IRD and Sobonne University) and the company Quinten (Paris France). Without the support and collaboration of some people, this thesis could not have been completed. Thus, I would like to thank all those who have contributed in any way to the progress of my work.

I had the chance to be supervised and directed by four exceptional people, gifted with great intelligence, kindness and benevolence, thanks to whom I was able to develop myself and complete this thesis in the best conditions. I would like to express my deepest gratitude to my thesis director Jean-Daniel Zucker, to my co-director Karine Clément and to my supervisor Edi Prifti for their precious help and their numerous advices since the elaboration of the thesis subject until the end of my PhD. I would also like to thank my thesis director at Quinten, Alexandre Templier, who trusted me and accepted to support my PhD and for its involvement in this adventure.

I am deeply grateful to Professors Jean-Philippe Vert and Edoardo Pasolli for having accepted to be the reviewers of my work. My warm thanks are also addressed to Professors Alessandra Carbone and Blaise Hanczar who agreed to examine this work and to participate in the thesis jury.

I thank Sorbonne University for allowing me to teach computer science subjects, for providing me with computational resources to run my programs and for offering me enriching trainings throughout my thesis. I also thank UMMISCO and Quinten for their welcome in their different teams and for the availability of their infrastructures. I would like to thank my professors at Sorbonne University who gave me a taste for computer science in my Bachelor's degree and for data science in my Master's degree. I particularly underline the work of the directors of the Master DAC Ludovic Denoyer and Bernd Amann, who proposed rich, relevant and interesting teachings, giving me the desire to continue in the field of Data Science and Machine Learning.

I would like to express my gratitude to my colleagues and friends at UMMISCO and Quinten for their indispensable help in the completion of my thesis. I thank Minh Dao Quang for his explanations on the use of Spark and his time spent on the development of a computational infrastructure in the laboratory. I thank Eugeni

Belda, Paul Deveau and Ari Ugarte for their explanations on bioinformatics and metagenomic analysis. I thank Cyril Esnault, May-Line Gadonna and Pauline Guilmin for their collaboration on the subgroup discovery approach. Finally, I also thank Mathilde Berthelot, Christelle Pezzucchi and Asya Grechka who helped me during the writing/proofreading of my thesis.

Abstract

Thesis title: End-to-End Deep Learning and Subgroup discovery approaches to learn from metagenomics data

Key words: Metagenomics, deep learning, embeddings, point-of-care processing, subgroup discovery, precision medicine, phenotype prediction

Over the past decade, technological advances have made high-speed, high-resolution sequencing of genetic material possible at ever lower cost (from millions to one hundred dollars). In this context, the human microbiome has demonstrated its great capacity to stratify and classify various human diseases, and is increasingly considered as our second "genome". As a "super-integrator" of patient status, the gut microbiota is set to play a key role in precision medicine. Omics biomarkers identification has become a major goal of metagenomics processing, as it allows to understand the microbial diversities that induce the patient stratification. There remain many challenges associated with mainstream metagenomics pipelines that are both time consuming and not stand-alone. This prevents metagenomics to be used as "point-of-care" solutions especially in resource-limited or remote locations. Indeed, state-of-the-art approach to learning from metagenomics data still relies on tedious and computationally heavy projections of the sequence data against very large genomic reference catalogs. In this thesis, we address this issue by training deep neural networks directly from raw sequencing data building an embedding of metagenomes called Metagenome2Vec. It supports learning models that perform accurate and fast stand-alone classification. Learning DNA embeddings is achieved with a reference catalog of metagenomic species used as input of a metagenome simulator. We also explore subgroup discovery algorithms that we adapt to build a classifier with a reject option which then delegates samples, not belonging to any subgroup, to a supervised algorithm. This approach leverages the strengths of both subgroup discovery and classification concepts creating an explainable stratification of the patients groups. Several data sets are used in the experiments to discriminate patients based on different diseases (colorectal cancer, cirrhosis, diabetes, obesity) from the NCBI public repository. We have also developed different models using a simulator of metagenomic reads corresponding to binary class disease states in order to perform an intrinsic and extrinsic evaluation of the different learning steps of our algorithm. Intrinsic evaluation was performed primarily in the metagenome embedding creation part to verify that the learned embeddings were consistent with the DNA chain distance scores. The extrinsic evaluation validated that the algorithms

correctly addressed the stratification problem and that the subgroup discovery part generates robust and credible metagenomic signatures. These evaluations show that our two methods reach high performance comparable to the state-of-the-art approaches, while being respectively stand-alone and interpretable. They are a proof-of-concept that pave the way for future "point-of-care" precision medicine based on metagenomics.

Résumé

Titre de la thèse: Approches basées sur les réseau de neurones et la découverte de sous-groupes pour l'apprentissage machine à partir à partir de données métagénomiques

Mots-clés: Métagénomique, apprentissage machine, apprentissage profond, découverte de sous-groupes, médecine de précision, prédiction de phénotype

Au cours de la dernière décennie, l'avancée technologique a rendu possible le séquençage à grande vitesse et à haute résolution du matériel génétique à un coût toujours plus faible (de plusieurs millions à une centaine de dollars). Dans ce contexte, le microbiome humain a démontré sa capacité à stratifier et à classer diverses maladies humaines, et est de plus en plus considéré comme notre deuxième "génome". En tant que "super-intégrateur" du statut du patient, le microbiote intestinal est appelé à jouer un rôle clé dans la médecine de précision. L'identification de biomarqueurs omiques est devenue un objectif majeur en métagénomique, car elle permet de comprendre les diversités microbiennes qui induisent la stratification des patients. Il reste de nombreux défis associés aux pipelines de métagénomique courants, qui prennent du temps et ne sont pas autonomes. Cela empêche l'utilisation de la métagénomique comme solution "point-of-care", en particulier dans les régions éloignées ou avec des ressources limitées. En effet, l'état de l'art de l'apprentissage à partir de données métagénomiques repose sur des projections fastidieuses et lourdes en termes de calcul des données de séquence par rapport à de très grands catalogues génomiques de référence. Dans cette thèse, nous abordons ce problème en formant des réseaux neuronaux profonds directement à partir de données de séquençage brutes, en construisant un encastrement de métagénomes appelé Metagenome2Vec. Il définit des modèles d'apprentissage qui effectuent une classification autonome précise et rapide. L'apprentissage des encastrements d'ADN est réalisé à l'aide d'un catalogue de référence d'espèces métagénomiques utilisé comme entrée d'un simulateur de métagénome. Nous explorons également des algorithmes de découverte de sous-groupes que nous adaptons pour construire un classifieur avec une option de rejet qui délègue ensuite les échantillons n'appartenant à aucun sous-groupe à un algorithme supervisé. Cette approche exploite les forces des concepts de découverte de sous-groupes et de classification, créant ainsi une stratification explicable des groupes de patients. Plusieurs ensembles de données sont utilisés dans les expériences pour discriminer les patients en fonction de différentes maladies (cancer colorectal, cirrhose, diabète, obésité) à partir

du répertoire public NCBI. Nous avons également développé différents modèles en utilisant un simulateur de lectures métagénomiques correspondant à des états pathologiques de classe binaire afin d'effectuer une évaluation intrinsèque et extrinsèque des différentes étapes d'apprentissage de notre algorithme. L'évaluation intrinsèque a été réalisée principalement dans la partie de création d'encastrements de métagénomes afin de vérifier que les encastrements appris étaient cohérents avec les scores de distance des chaînes d'ADN. L'évaluation extrinsèque a permis de valider que les algorithmes abordent correctement le problème de la stratification et que la partie de découverte des sous-groupes génère des signatures métagénomiques robustes et crédibles. Ces évaluations montrent que nos deux méthodes atteignent des performances élevées comparables aux approches de l'état de l'art, tout en étant respectivement autonomes et interprétables. Elles constituent une preuve de concept qui ouvre la voie à une future médecine de précision "point-of-care" basée sur la métagénomique.

Table of Contents

1 Introduction

1.1 Background and rationale

1.1.1 Context

Our vision of the human being seen as an entity in itself has evolved to leave room for a more global vision where the ecosystems that we shelter are taken into account. The term “holobiont” is used to define this concept and characterizes a host (a person, an animal, a plant...) with all the microorganisms living inside. Indeed, the role of the microbiome for its host is so important for physiology, psychology, health that it is relevant to considered them together [Ber+20]. Health practitioners are increasingly using diagnostics based on new accessible “big data” such as clinical, environmental, “omics” (genomics, transcriptomics, radiomics, etc.), including so-called metagenomic data, which is the quantifying of the metagenome, a “super-integrator” of patient’s environment and lifestyle impacting its condition. This health care is part of the field of precision medicine which is opposed to the “one-size-fit-all” vision where all patients receive the same adapted treatments (e.g., dosage) and improving the health of 4-25 % of the population [Pet18]. Precision medicine favors the discovery of specific characteristics and treatment personalization for each individual or subgroup of individuals. The integration of “big data” for diagnoses in precision medicine represents a major challenge in terms of analytical complexity and large computing volumetry. Data science and artificial intelligence play an important role in this field and are subject to active research to address these challenges.

This thesis is the result of the collaboration between two laboratories (UMMISCO (Unité de Modélisation Mathématique et Informatique des Systèmes Complexes) and Nutriomics) and a data science company (Quinten). UMMISCO is an international mixed lab developing computer and mathematical modeling methods for “complex system”. Nutriomics concentrates its research on the gut microbiota, intestine, adipose tissue remodeling and systems’ biology with systemic approaches and “big data” integration. Finally, Quinten is a consulting company that has built up an analyt-

ical knowledge applied to health on various subjects and with different partners, integrating care and precision medicine solutions in real-world set up.

Our researches focus on the manipulation and analysis of metagenomic data aiming at a “point-of-care” processing and phenotypic signatures identification of medical interest related to cardiometabolic diseases. “Point-of-care” is a medical laboratory diagnostic intended to be performed in close proximity to the patient and get results in real time like a few hours and not many days, a challenge for current metagenomic workflow.

1.1.2 Exploring microbial environments with metagenomics

1.1.2.1. An overview of the microbiota

Microbiota is a term to define the set of microorganisms (bacteria, viruses, fungi, yeasts) living in a specific environment called microbiome. These microorganisms are present almost everywhere in our bodies, interacting with each other and with their environment. For a long time, these microorganisms were suspected of being responsible for diseases causing epidemics or even pandemics, but it is only since the end of the 19th century with the arrival of the microscope that this relationship has been demonstrated. More recently, some studies revealed that these microorganisms have also a curative power to treat human diseases [Cha+20]. This leads to the analysis of the functionality and impact of microorganisms on the human phenotype.

It is estimated that there are about one to two times as many microbial cells in the human body as there are human cells [SM16]. Furthermore, the microbiome is constituted by several million non redundant genes compared to “only” about 23,000 for the human genome. The genome is the entire genetic material of an organism while genes are regions of genomes that encode for macro molecules called proteins with a wide range of functions in the body. The microbiome was later shown to play a crucial role not only in the environmental ecosystems but also in relation with the host they inhabit. That is why with these characteristics, Zhao [Zha10] states that humans have two genomes, their own and one of their microbiota made up of microorganisms acquired from the environment. When the human gut microbiome is altered (it is the largest reservoir of bacteria that inhabits them and can reach several kilograms), it often results in impaired human health. Indeed, recent research has demonstrated the strong relationship between these

microorganisms and complex and chronic human diseases such as diabetes, cirrhosis, autism, obesity or cancer [Lia+18; Wir+19].

The microorganisms inside an ecosystem can form inter-member organizations, sometimes referred to as "guilds", and create localized interactions that influence their ecosystem [Wu+21]. It is thus essential to analyze the composition and the interactions of microbiota members when we study them. Metagenomics manages these objectives and refers to the set of methods supporting microbiota by sequencing and analyzing DNA of several individuals of different species in the same environment. Although in theory the metagenomics can be used for all kinds of organisms, it is mostly used to explore the structure of microbial communities living in a given ecosystem.

Over the past decade, technological advances have made high-speed, high-resolution sequencing of genetic material possible at ever lower cost[1], from millions to one hundred dollars Wetterstrand [Wet20]. Such improvements have allowed a whole field - that of metagenomics - to develop and maturing very quickly with large public repositories increasing the standardized dataset [OC15; GMM16; Mar17].

1.1.3 Metagenomics in precision medicine

In modern medicine, targeting certain patient populations that would benefit from a particular treatment is becoming an important goal [LCZ19; Kor18]. Precision medicine is defined as the tailoring of medical treatments to the characteristics of individuals, classifying them into sub-populations that differ in their response to a disease or their response to a treatment. Precision medicine therefore aims to create models based on the analysis of data characterizing patients. The objective is to propose more effective therapeutic solutions by acting specifically on the potential causes of the disease. Since it has come to the forefront of patient treatment, personalized data such as electronic medical records or DNA sequencing have proliferated, enabling the development of many techniques and treatments [GP18].

Metagenomics has become a major area of research in precision medicine [Job18] and is of great interest in personalized treatments mainly for two reasons: Microbiomes define characteristics specific to each individual at the same level as their own genetic material, and recently, sequencing data has increased exponentially, as explained in section 1.1.2.1. That is why metagenomic analysis is rapidly moving from research to clinical laboratories to attempt to counter both infectious

[1] The decay far exceeds Moor's empirical law which states that the information power doubles every two years.

[CM19] or chronic [NF18] diseases and to guide patients on special diets and other interventions modifying their intestinal flora [Lee+21].

1.1.4 Overview of different sequencing technologies

Metagenomic data collection is performed using Next-generation high-throughput DNA sequencing technologies (NGS) allowing to sequence the DNA of any organism providing precise biological information. These methods are parallelized, millions of reactions take place at the same time in flow cells that contain fiber optic wells which generate millions of sequence reads in a short time (see Figure 1.1). Reads are short sequences, generally between 50 to several thousands, of base pair Adenine (A), Cytosine (C), Guanine (G) and Thymine (T), also called called nucleotides, and are stored in standardized files in fastq format.

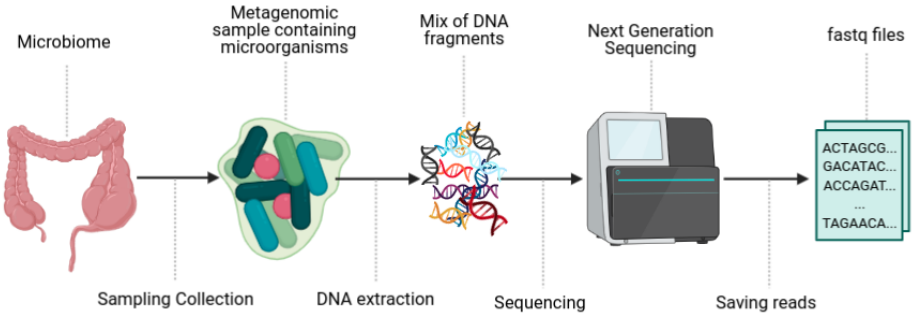


Fig. 1.1.: Example of illumina sequencing from the intestine: stools (representing the microbiome) are collected, the DNA of the microorganisms is then extracted to be passed in an illumina NGS which will sequence this DNA and save it in fastq files.

1.1.4.1. Illumina Technologies

The most used NGS technology is illumina [Hua+12] (about 56% of the market) and the data manipulated during this thesis mainly come from this technology. Illumina allows identifying simultaneously the DNA bases when they are incorporated in the nucleotide chain. Each base emits a unique fluorescence signal when added to the strand being synthesized and it determines the DNA sequence. The technology has a low insertion/deletion error rate, but the size of the fragments does not exceed three hundred base pairs which induces a very high number of sequenced fragments adding more difficulties for assembling the fragments into contigs. (see Figure 1.2 for more details). However, these technologies are not portable, do not produce real

time data and need important preparation steps. All these aspects are limitations restricting the possibilities of metagenomic sequencing analysis and usages.

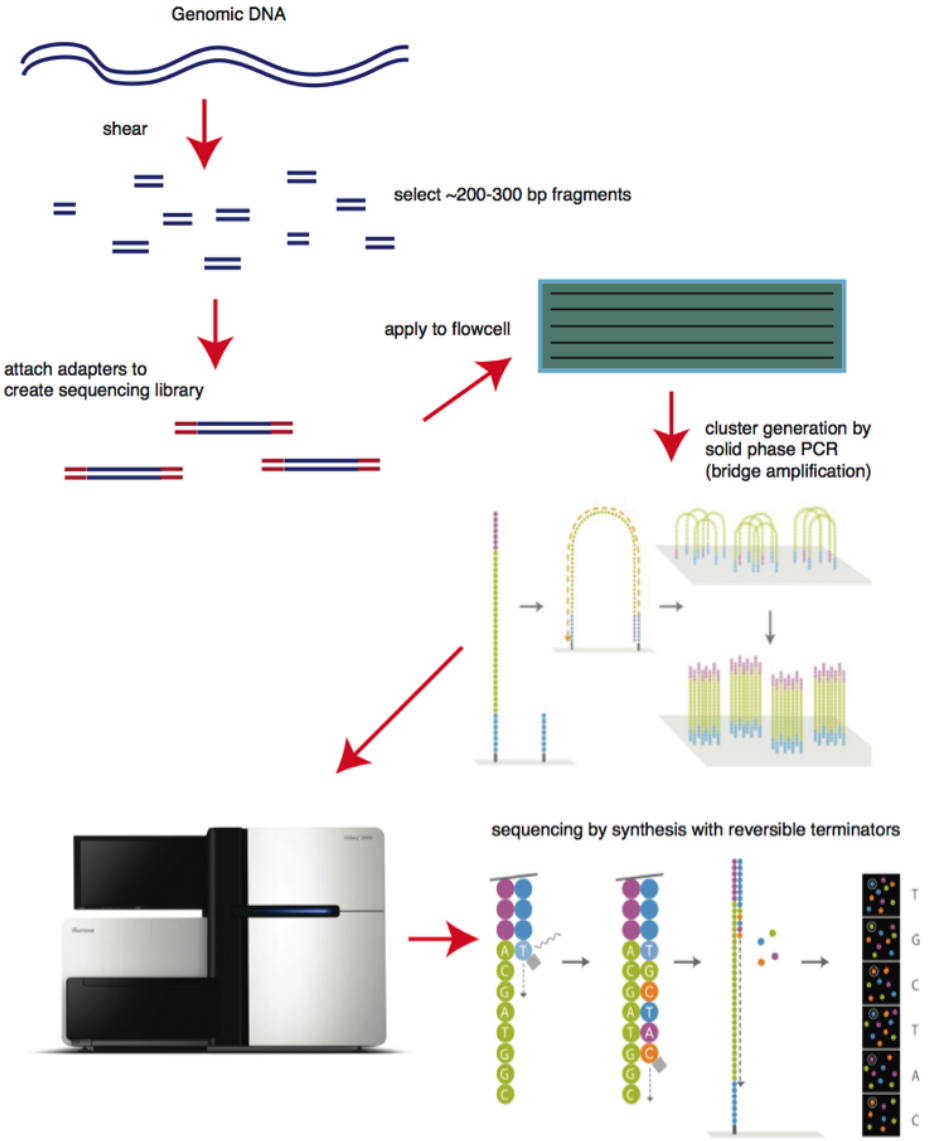


Fig. 1.2.: Illustration of the Illumina sequencing. The genomic DNA is cut in small fragments of $\simeq 200 basepair$ where adapters are attached to create sequencing libraries. The libraries are flowed on a solid surface where the fragments bind and then are amplified using clonal amplification and Polymerase Chain Reaction (PCR) methods to generate clusters. This results in around one million copies of each sample on the flowcell surface before to be sequenced by synthesis producing the DNA reads. Image credit: [Bro12]

1.1.4.2. Oxford Nanopore Technologies (ONT, third generation)

The so-called third generation sequencing technologies has been developed by Oxford Nanopore Technologies Ltd [Jai+16] and allows the acquisition of long reads from 10^4 to 10^6 base pairs. The technology does not need PCR amplification or chemical labeling of the sample to sequence molecule of DNA or RNA (see Figure 1.3 for more details). Long reads can overlap long repeats of DNA fragments and thus contigs orientation are less ambiguous, which is essential for *de novo* genome assembly. However, third generation sequencing technologies currently have a high error rate because they do not use a cyclic method (addition of the nucleotides in a cyclic way one by one, always in the same order and successively). Indeed, the DNA molecule is decrypted in real time by a high frequency detection method. In our experiments we did not use public data acquired by ONT, however we simulated metagenomic data with the CAMISIM software [Fri+19] combined with NanoSim [Yan+17] to evaluate if our approaches could also be effective on these data (see section 2.2.2). We are interested in manipulating such data because these are the sequencing technologies that could be used most in the future in a precision medicine context. Furthermore, these are in line with our approach which consists in designing a "point-of-care" solution especially especially in resource-limited or remote locations.

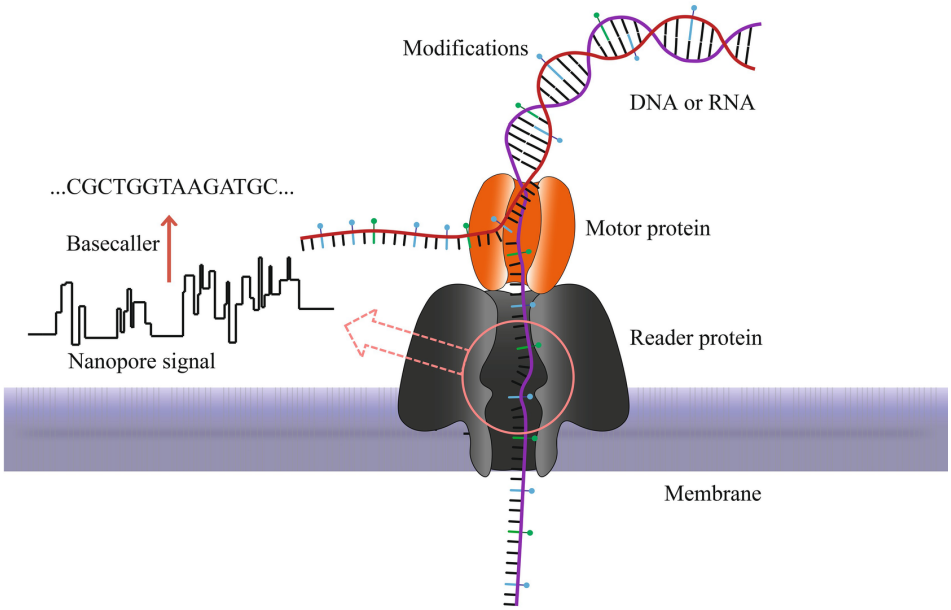


Fig. 1.3.: Illustration of the Nanopore sequencing. The motor protein passes the nucleic acid molecules (DNA or RNA strand) through the nanoscale pore provided by the reader protein. This causes current fluctuations in the membrane whose signal is converted with the corresponding nucleic acid sequence. Image credit: [HCR21]

1.1.5 Bioinformatics workflows to analyze metagenomic data

The development of metagenomic sequencing came along with the rapid development of bioinformatics workflows, which ultimately yield quantitative measurements of biological objects such as genes, species, genera and other taxonomic levels, functional pathways, etc in the form of relative abundance tables [Kun+08; NP16; Wen+17; Qui+17]. Several steps are required to obtain such count tables and all of them rely on assumptions that affect the final outcome. The complex bioinformatics workflow starts by reading the fastq files and use quality scores to filter out nucleotides as well as reads that do not pass the defined confidence criteria. Next, the reads can be aligned onto the host genome or assembled to form longer sequences called contigs while removing redundant one. Finally, the resulting reads are grouped together (binned) using different techniques, including alignment with reference gene/genome catalogs or through other approaches based on k-mer similarity or co-abundance clustering [Met+14b; QC19]. A condensed view of the workflow is illustrated on Figure 1.4. After these bioinformatics processing steps, analysis results in several types of applications such as pathogenicity prediction, biomarker discovery, species interaction analysis, classification / clustering, phylogeny reconstruction, epidemiology, clinical interpretation or epidemiology [Qui+17]. For phenotypic or pathoneginc prediction, a taxa abundance table is extracted from the metagenomic workflow and is then handled by state-of-the-art classifier models (*SVM*, *Random Forest*, *Penalized Logistic Regression*, etc ...) for further statistical analyses. This section highlights the main stages of the bioinformatic workflow and their different characteristics.

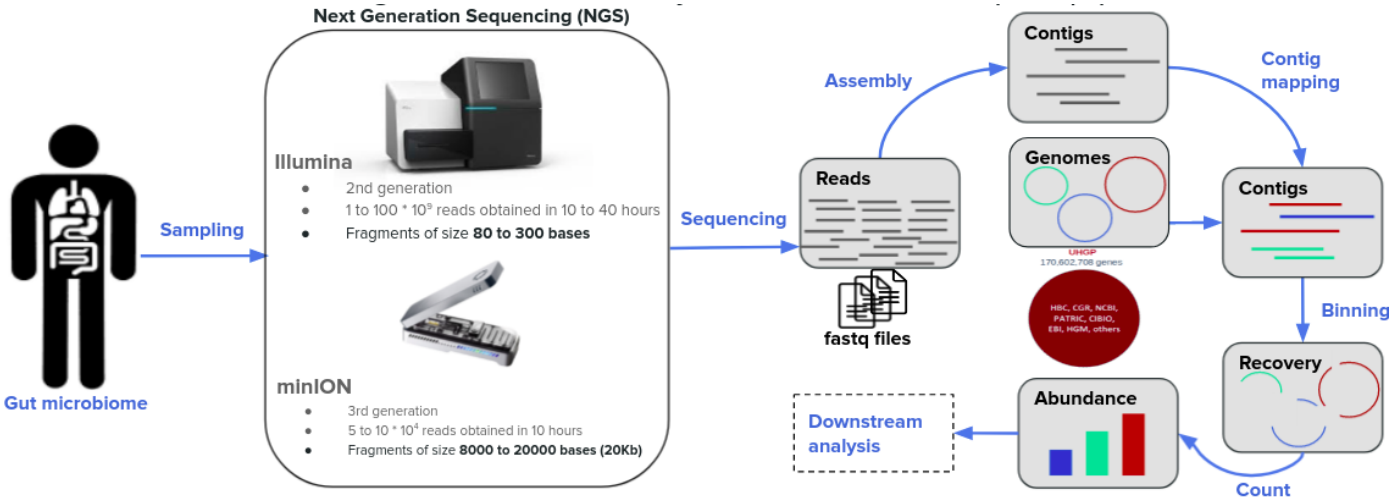


Fig. 1.4.: Schema of a Bioinformatics workflow processing metagenomic data. After a microbiota has been sequenced by NGS, the fastq files are cleaned, then all the reads are assembled into contigs, forming bigger sequences that are mapped on reference catalog and then binned to individual genome to recover the number of taxa present in the initial microbiome.

1.1.5.1. The importance of reference catalogs

The National Center for Biotechnology information (NCBI) lists a large number of genomes representing about 10% of living species. We are therefore far from covering all microbial genomes, which prevents us from classifying all their genes to constitute the pangenome, describing the full range of genes of each species. To circumvent the availability of genomes, thanks to NGS, the MetaHIT consortium was able to build over the last few years a catalog (integrated gene catalog (IGC)) of 9.9 million non-redundant genes from shotgun sequencing of fecal samples organized in "metagenomic species" (MGS) [Met+14a]. This is called *de novo* sequencing, a reference-free technique to discover and reconstitute gene repertories of microbial species. The method sequences novel genome without reference sequence for alignment by assembling reads as contigs [Met+14b; Pla+19]. A recent study has unified more than 200k reference genomes from the human gut microbiome to create the Unified Human Gastointestinal Protein (UHGP) catalog. Unlike IGC, this catalog provides links between genes and their genome necessary for taxonomic classification, establishing genetic relationships and inferring complete functional pathways on a genomic basis [Alm+21]. These catalogs of genomes or genes are references that allow bioinformatics pipelines to perform assembly or alignment tasks to quantify the microorganisms in the microbiota. Thus, most of the downstream analyses depend on the catalog choice.

1.1.5.2. Preprocessing and sequences alignments to compute abundance tables

The bioinformatic workflow composed of different stages is assembled to analyze metagenomic data. This is called quantitative metagenomics and it aims to measure biological objects from sequences to build abundance tables (Figure 1.5). Assigning MGS to each read or contig, called the binning process, is one of the main objectives and one of the most challenging of metagenomic analysis. Preprocessing of the metagenomic data is performed on the raw fastq files output from the sequencer. A cleaning stage removes reads that are too short, have a low-quality score or come from human. Next, two distinct methods can be applied: read-based or and assembly-based [Har+19].

Read-based uses the *close-reference* strategy that consists in clustering the reads against a reference catalog (collection of sequences) [Met+14b], if a read cannot be mapped it is excluded from downstream analyses. Read-based strategy is fast but it ignores sequences that are not in the catalog databases.

Assembly-based method uses the de novo strategy (seen in section 1.1.5.1). The reads are compared to each other to assemble them into contigs to form a consensus sequence that can be annotated on a database or represent an unknown species. In that way all reads are clustered, but the calculation time is increased. Reference-free methods often are interested in the use of k-mer count [PPV19; Aud+17]. These methods measure statistical information about the sequences by projecting them into a k-mer feature space, which allows us to compute distances between sequences to find the most similar pairs. [Zie+19].

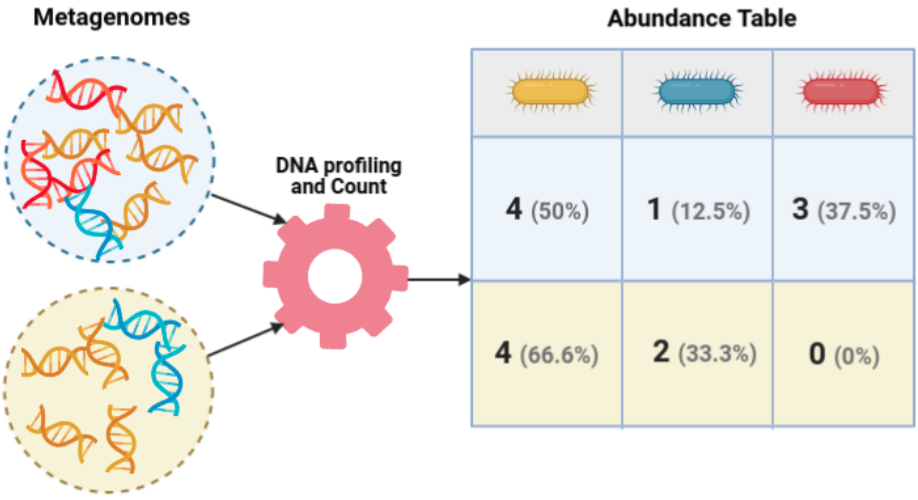


Fig. 1.5.: An example of an abundance table where two metagenomes have different numbers of species. For yellow DNA, both have an absolute abundance equal to four, but the relative abundances in percentage are different: 50% for the former versus 66.6% for the latter. Relative abundance is expressed as a percentage and thus provides the proportion of one species to the others.

1.1.5.3. Data analysis from metagenomic data

Metagenomics revolutionized the analysis of microbial ecology. Many challenges emerge from this field, notably due to the complexity and large volume of data. From the sequencing of genetic material, through data processing, to the different possible analyses mentioned in section 1.1.5, many methods have been developed to reduce the computation time, increase the reliability of the analyses, improve the results or even answer new problems [PGB20]. In this section we explain the different possible metagenomic applications in terms of data sequencing and objectives to situate the orientation of the thesis.

Sequencing methods There are two main sequencing methods in metagenomics to extract information about microorganisms (see Figure 1.6).

- *Amplicon sequencing*: It sequences the rRNA or ribosomal DNA of organisms. It consists in sequencing a unique gene and not the whole genome. This gene must be common to several species while presenting sufficiently variable regions to discriminate a species. It can be the 16S rRNA gene for bacteria / archaea or 18S rRNA gene for eukaryotes.

- *Shotgun sequencing*: It sequences full genomes (i.e. all genes present) of the microorganisms in the environment with high-throughput sequencers.

Shotgun sequencing is thus more precise than amplicon sequencing because it does not focus on a specific gene allowing for example to describe the global functioning of the microbiota. Nevertheless, this makes the algorithms for processing the data more complicated because there must be an additional step of assembling the genomes. The data manipulated during this thesis are from *shotgun sequencing*.

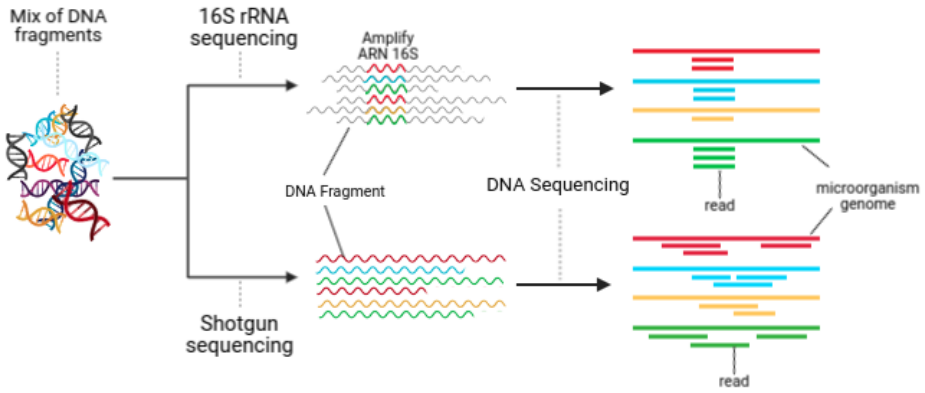


Fig. 1.6.: Shotgun vs 16S sequencing: 16S rRNA sequencing will focus on the sequencing of a single part of the genome common to each species. As a result, the reads will align to the same location on the genome part. For the shotgun method the whole genome is considered which will produce reads that can represent any part of the genome.

Objectives We can distinguish two objectives in metagenomics. The *sequence-based metagenomics* which determines the provenance of a sequence, and answers the question "from which taxon does the sequence originate ?". This gives an estimate of the proportion of microorganisms in the environment. The other is the *function-based metagenomics* which measures the expression of genes and answers the question "what are the functions of the genes ?". The goal is here to understand how bacterial communities interact and what are their roles in the environment. In this thesis, we focus on the first objective by exploiting metagenomic composition to address the problem of phenotypic classification.

1.1.6 Classification models in metagenomics

1.1.6.1. Machine Learning on metagenomic data from gut microbiota

The huge computational steps for preprocessing several terabytes of metagenomics data (for most datasets) produce taxonomic abundance tables. The rows and the columns represent N samples (patients, environments, ...) and D features (species, genus, ...) respectively with the distinction of having $D >> N$. This is due to the fact that, even today, few samples (a few hundred) are available per dataset, making their processing a computer challenge. Several studies applied statistical analyses on these abundance tables as biomarker discovery, species interaction clustering or patient classification [Qui+17]. Biomarker discovery is an important challenge in medicine because this has the goal to yield meaningful biological information. In the case of metagenomics, biomarker discovery solves the problem of finding which microorganisms are likely to explain the difference between certain samples.

Segata et al. [Seg+11] proposed a method named *LefSe* (for linear discriminant analysis (LDA) effect size) applied to high-dimensional data for biomarker discovery to identify genomic features that distinguish sample classes. Some recent methods based on microbial ecosystem interactions address this problem. Prifti et al. [Pri+20] proposed the *Predomics* approach, a family of classification algorithms which uncover and explore biomarkers and are a simplification of linear models to be even more interpretable. Another approach, called *GutBalance*, [YZG21] uses discriminative balance analysis (DBA) method in order to select distal balances of pairs and trios of bacteria. Other studies have focused on supervised learning by comparing several state-of-the-art algorithms to assess the strength of microbiome-phenotype associations by evaluating the generalization of disease-predictive models across cohorts [EDF15; Pas+16; OZ20]. Thomas et al. [Tho+19] also analyzed, through several studies gathering 969 fecal metagenomes, the reproducibility of metagenomic biomarkers potentially linked to colorectal cancer with random effect models on the microbial richness and diversity. Harris et al. [Har+19] were interested in the results obtained by classifiers on quantitative metagenomic data constructed from the different approaches cited above namely read-based taxonomy profiling and assembly-based method. They found that there was not a large difference between the two approaches, the random forest model achieving the best results in both cases. Moreover, some R packages have been created like *MegaR* [Dhu+21] to propose a pipeline of metagenomic phenotype classification including model fine tuning, data processing, multiple machine learning (ML) techniques, model validation, and sample classification.

Finally, rather than using quantitative metagenomics, algorithms have been developed to manipulate k-mers directly from fastq files which has the benefice to be a reference-free method. K-mers are composed of k nucleotides and refer to all sub-sequences from a read obtained by DNA sequencing. The possible amount of k-mers given a read of length L is $L-k+1$, and the possible number of combinations is equal to 4^k (since there are 4 distinct nucleotides). K-mers count tables used as input of classifiers are created using specific optimization like Bayes classifier to filter and remove non-relevant k-mers [Lor+20; Ngu+].

One of the major difficulties often underestimated is the composition of the quantitative metagenomic data. Indeed, the number of sequences generated (sequencing depth) by NGS is not the same and varies from one sample or study to another. When the biological objects included in the samples are counted, it should not be restricted to an absolute count because it would not be representative of the real composition. A normalization step must therefore be applied and consist of dividing each abundance by the total number of taxonomic units, resulting in a table of relative abundance (see Figure 1.5). These data are defined as "compositional data" that mathematically represent points on a simplex. They provide relative information with quantitative descriptions of a set. The relative abundance, namely the percentage of total abundance, restrict the data to a sample space with the constraints of having the sum of each characteristic always equal to 1 and having their values included in the interval $[0, 1]$. These constraints require specific mathematical transformations to avoid misinterpretations or irreproducible analyses [YZG21]. The data processing often used are log-ratio transformations and refer to Additive Log-Ratio (ALR), Centered Log-Ratio (CLR) and Isometric Log-Ratio (ILR). The choice of the method is defined by the desired interpretation:

- ALR: Isomorphic[2] but not isometric[3]. Transforms the original D features to $D-1$ features space.
  Formula:
  $$alr(x) = [ln\frac{x_1}{x_D}, ln\frac{x_2}{x_D}, ..., ln\frac{x_{D-1}}{x_D}]$$
- CLR: Both isometric and isomorphic. It removes the value-range restriction, but it does not remove the sum constraint. It does not change the dimension of the basis as the ALR or ILR making it easier to train interpretable models
  Formula:
  $$clr(x) = [ln\frac{x_1}{g(x)}, ln\frac{x_2}{g(x)}, ..., ln\frac{x_{D-1}}{g(x)}$$

[2]Isomorphic: meaning that the mapping between the simplex and the new basis is preserved
[3]Isometric: meaning that the distances in the simplex are equivalent to the distances of the new transformed values

Where g(x) is the geometric mean of x.

- ILR: Isomorphic and isometric. It is often the most suitable transformation that manage the issue of sum and range value constraints because it is associated with orthonormal bases in the simplex. Nevetheless, as ALR it transforms the original D features to $D-1$ features space.
  Formula:
  $$ilr(x) = clr(x) \cdot \Psi'$$
  $$\Psi\Psi' = I_{D-1}$$
  Where Ψ is a $(D-1, D)$-matrix whose rows are $clr(e_i)$ and $e_1, e_2, ..., e_{D-1}$ is a generic orthonormal basis of the simplex S^D.

As ILR transformations are difficult to interpret, recent studies have defined a method called balance [QE20; YZG21], which is the log-ratio of the geometric means of two non-overlapping groups of features defined by a sequential binary partition (SBP). In that way, balances are more interpretable than common log-ratio transformations. Metagenomic compositionality is also managed by Friedman and Alm [FA12] to create a clustering graph network interaction of species. They proposed a robust approximation method called *SparCC* to derive the correlation matrix based on a rough estimate of the variance of the ratio-log of species.

1.1.6.2. Difficulties and weaknesses

There are some limitations to manipulating metagenomic data for downstream analysis:

- Bioinformatics workflows depend on different softwares, which are not typically designed to work together in the most efficient way.
- The catalogs processed for alignment may change from study to study, resulting in some bias in the analysis.
- Many parameters and thresholds must be set, often arbitrarily, which affects the final result.
- Two to Four hours of computation on a cluster with 9 nodes containing 56 cores CPU and 256GB of RAM are required to predict the class of one sample [Uga+19] which is not compatible with “point-of-care” treatment.

- Metagenomic models are trained to find biomarkers or to perform a specific classification task but rarely at the same time which do not provide interpretable results. When state-of-the-art metagenomic methods manage this problem [Pri+20; YZG21], they create a global stratification with a unique model and do not provide a localized / personalized stratification per patient, an important concept for precision medicine.

All these difficulties are the area of improvement that we have addressed during this thesis. The proposed solutions are discussed first in section 1.2 and then detailed in the other chapters.

1.2 Research problem and contributions

1.2.1 Objectives

In this thesis, we focus on analyzing metagenomes to develop predictive and explainable models for a stratification at the phenotypic level. We aim to take the respective strengths of interpretable and black box models, and adapt them to the context of metagenomic analysis for precision medicine. The methods we have developed to address the different problems reported in section 1.1.6.2 are detailed in this section and summarized below:

- An end-to-end deep learning (DL) approach learns a compact representation of metagenomes by taking raw DNA sequences directly from NGS. The interest is to use metagenomics as a "point-of-care" solution especially in resource-limited or remote locations without the need to send the data for heavy processing to bioinformatics platforms.
- A subgroup discovery (SD) algorithm built as a classifier with a reject option. It delegates samples, not belonging to any subgroup, to a supervised algorithm. This leverages the strengths of the discovery and subgroup classification concepts by creating an explainable stratification of patient groups.

Machine learning (ML) approaches, and in particular deep learning, are more and more common in the field of metagenomics whatever the application. To improve their performance, ML models have seen their complexity increase over time. However, it is known that, in statistical learning, the complexity of the algorithms makes the interpretation of predictions difficult to explain [CPC19]. In many fields such as medicine, the transparency and confidence of the results often

are needed. Explicability brings a certain insight on the decision of the algorithm by limiting the potential bias that is often not obvious to detect and that is learned by a program on a dataset [Gil+19]. It enhances the fairness of the algorithm and the prediction. Medicine is a field where explainability plays a major role because errors can be serious for the health of patients and not being able to understand what the model indicates is a serious issue.

1.2.2 Deep learning based approach and point of care

DL has become very popular in the past decades thanks to the improvement of computer components, especially GPU cards that allow to greatly accelerate the speed of calculations. DL is currently the state of the art in almost all domains, especially with complex data such as text, image, video and sound [Alo+19]. More recently, several studies have used deep learning for bioinformatics applications and achieved excellent results such as in drug discovery [ZLX20], medical image classification [Ges+19] and even electrocardiogram diagnosis [Rib+20]. One of the strengths of DL is that it can learn complex object representations, known as *embeddings*, without extensive feature engineering. Embeddings are a mapping of discrete, categorical or continuous variables to a vector of continuous numbers. It encodes the meaning of an object in a learned vector space, making possible mathematical operations between objects such as comparing their distance or combining them in a certain way. Metagenomic data also have a complex structure that requires transformations to be learned by ML algorithms. Adaptations of existing DL models in Natural Language Processing (NLP) for metagenomics have rapidly emerged making DL a promising approach to process metagenomic data [Min+17; Wol+18; Roj+19; MV19; Lia+20; Geo+20].

One weakness of DL models is their complexity which tends to considerably reduce their interpretability. Black box models need post-hoc methods to bring interpretability to the result [Mol21]. These methods differ in purpose, they can be global to explain the whole model or local to explain an individual prediction. There are several recent techniques that have been developed to address this issue like *Interpretable Local Surrogates*, *Occlusion Analysis*, *Gradient-based techniques*, *Layer-Wise Relevance Propagation* [Sam+20]. Koras et al. [Kor+21] proposed a tailored interpretability method assigning a biological meaning to the individual dimensions of the hidden space in a problem of drug sensitivity prediction for cancer. Another approach is to add an attention mechanism to train neural networks [LPM15; BCB16] which helps the model to select and concentrate only on the most interesting features. Attention can be used to identify information that models find important like

a region in an image or a word in a sentence. However, all of these approaches have limitations because they are only an approximation of interpretation.

1.2.2.1. Similarities and differences between DNA sequences and Natural Language Processing

Raw data from NGS are not structured data and must be converted to a suitable format (e.g. tabular) to solve problems related to metagenomics by applying machine learning algorithms. This is the case of bioinfomatics workflows described in the section 1.1.5 which produce taxa abundance tables. To reduce the complexity of the conversion, a strategy can consist in using a DL model trained on this type of data which automatically learns an adapted representation. In fact, DL has been declined in different structures specific to certain use cases such as recurrent neural networks (RNN) for sequence, text and audio analysis or convolution neural networks (CNN) for image analysis. In recent years, distributed representations of words in a vector space have been increasingly used in NLP to improve the performance of learning algorithms [Mik+13]. These representations are *embeddings*, characterizing words in a numerical vector space capturing semantic and lexical information learned with contexts of words. These vectors can thus be used in many applications like sentiment analysis [Maa+11], translation [Qi+18] or even speech recognition [BH14], outperforming standard word count representation. To go further, considering the concept of word embeddings, it is possible in NLP to build representations of sentences or documents with different techniques [KW14; SVL14; HCK16; Dev+18]. In metagenomics, DNA sequences can be embedded in an identical way with some preprocessing [Ng17; MV19; Jou+16]. A DNA sequence is composed of four nucleotides A, C, G and T. Therefore, it can be similar to a natural language sentence with the difference that it is based on a shorter alphabet. However, it is necessary to take into consideration other distinctions between metagenomes and NLP data. DNA sequences do not have an explicit concept of words delimited by spaces between letters. DNA sub-strings k-mers are generally considered. Furthermore, NGS makes "massively parallel sequencing" to numerically convert several DNA fragments into short reads. Thus, metagenomic data is composed of several sequences without any information about the order; a drastic change from textual data in which sentences and paragraphs should be read in a successive manner. The Figure 1.7 makes an analogy between NLP and metagenomics.

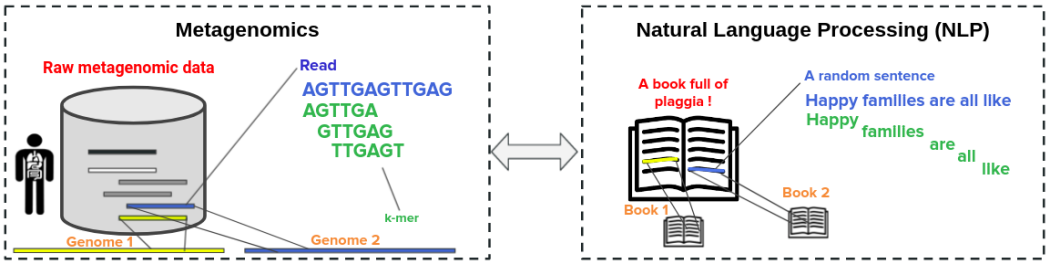


Fig. 1.7.: Analogy between Metagenomics and Natural Language Processing. A metagenome is composed of several copies of genomes, which can be similar to a book full of plagia in which there are several sentences of different books. As a sentence is composed of words and is a part of a book, we can match this with a read cut into k-mers corresponding to part of a genome.

1.2.2.2. Our approach: *Metagenome2Vec*

In this thesis, we bypass the classical bioinformatics workflow and instead aim to directly classify metagenomic samples from raw sequence data in an end-to-end process [Que+21]. Moreover, the resulting trained DNN can even automatically discover important biological concepts responsible for the classification. Using such a framework can solve the bottleneck of data insufficiency present in classical methods. Moreover, real-time results could be given with our proposed framework, especially with the improvement of NGS technology such as Nanopore.

Method: The core of our approach lies on the integration of different types of embeddings that encode the metagenomic sequences. We divide this pipeline into four main stages and assign a name to each of them for more clarity. The first one, *kmer2vec*, consists of a transformation of k-mers into embeddings. The second, *read2vec*, refers to reads projection into embeddings. *kmer2vec* and *read2vec* act as NLP models that transform words and sentences into vectors. The third, *read2genome*, classifies reads into bacterial genomes from which they most likely originated. The goal of this step is to estimate the abundance of taxa present in the metagenomes, thus allowing embeddings of reads from the same class to be grouped together in order to amplify their information. The fourth and last step, *Metagenome2Vec*, begins by transforming the metagenomes into robust multiple instance representations using *read2vec* and *read2genome* and drastically reduces the initial dimensional complexity. Finally, two DL models are implemented to handle multiple instances is trained on the transformed metagenomes' vector space. One architecture is a Variational Auto-Encoder [KW14] to encode all the information in a single vector. The second DL model, called deepSets [Zah+17], is trained to

classify the labels of the metagenomes from the classes to be discriminated. The entire pipeline and concepts are detailed in chapter 3.

Results: *Metagenome2Vec* achieves good results by exceeding the state of the art on several datasets and on various evaluation metrics. The abundance of metagenomic species calculated by the deep learning model is often sufficient to get the best scores without the final use of embeddings. On the other hand, it turned out that on a dataset of simulated Nanopore reads, it was better to use the embeddings to maximize our scores. Finally, the best option is to combine the calculated species abundance with the learned read embeddings, which allows us to keep the best scores no matter which approach performs better initially.

Metagenome2Vec is a black box algorithm based on deep learning. Despite its good performances and its ability to make "point-of-care" processing with an inference time about 1 hours per sample with 24-48 CPUs, it does not provide a relevant understanding of the predictions. The next section illustrates a second method studied, based on the subgroup discovery, to alleviate the interpretability problem and be in line with precision medicine.

1.2.3 Building interepretable signatures based on Subgroup Discovery

1.2.3.1. Introduction of subgroup analysis

Searching for subgroups of items with properties that differentiate them from others is a very general task in data analysis that refers to subgroup identification. There are a large number of methods for finding these subgroups that have been developed in different areas of research. Depending on the field of application, the algorithms considered differ in particular on the metrics used to qualify the groups of interest. The field of medicine is one with the most application for subgroups search. Indeed, the considerable heterogeneity in disease manifestation and response to treatment remains a major challenge in medicine. Understanding what drives such differences is critical to adjust treatment strategies, guide drug development, and gain insights into disease progression. A literature in ecology emphasizes the fact that several causes can lead to the same effect [Moo01]. This is illustrated by the notion of the "Karenine effect" [ZMV17], where several individual profiles with different characteristics can exist for the same phenotype.

Symbolic approaches are a field of machine learning that deals with the inductive learning of symbolic descriptions such as rules, decision trees or logical representations. [FGL14]. The two main branches from this category of algorithms are subgroup discovery and classification rules learning. A rule covers a subgroup of samples in the database that it characterizes by a certain value of the variable of interest. Subgroup discovery algorithms, on the first hand, create individual rules which are sub-signatures of the dataset describing the properties of individual groups specific to a target class. Classification rule learning algorithms, on the other hand, create a combination of rules covering the entire data set assigning a prediction to each example [NLW09; Val+17b]. SD is therefore more selective, it keeps only the rules that meet certain degrees of robustness and that are statistically credible. In the field of subgroup discovery there are two families of approaches: the first is Subgroup Identification (SI) and the second is Knowledge Discovery in Databases (KDD). SI is more specific to medical data analysis because the generation of subgroups is driven by both treatment arms and the outcome, while KDD is linked with data mining culture.

1.2.3.2. Subgroup Discovery

SD defines a category of models that provides interpretable patterns unlike well-known state-of-the art-ML algorithms (e.g., *SVM*, *Random Forest* or also *Neural Networks*) returning black box patterns [Imp12]. SD performs analyses where the goal is to capture knowledge through the data and not to make the model itself the source of knowledge. Although interpretability can be designed in black box models, notably by looking at the importance of the variables returned by the model or the weights learned, this does not decrease the complexity of the model. Molnar [Mol21] states that two kinds of interpretability methods have to be distinguished: intrinsic and post-hoc. SD is related with intrinsic interpretability because it has a simple structure naturally understandable, whereas post-hoc interpretability is related to black box models as they need interpretation methods after the training stage.

In bioinformatics, several studies aim to discover patterns based on interpretable models. This is the case of microarrays where the models seek to explain the expression level of a large number of genes simultaneously at a given time and in a given state compared to a reference sample [Li+03; NLW09]. As metagenomics collects the genetic material specific to each individual, which differs from other sources of data like gender, age or biomedical measures that could be similar from one patient to another, it thus supports precision medicine allowing to create

individualized treatments for each patient. SD is designed to create predictions that can be interpreted via simple formulas that enable to have leverage on the different characteristics present in these formulas. For precision medicine in metagenomics, SD could help decision making by providing results defining metagenomic profiles related to certain phenotypes, i.e. the impact of the bacterial balance regulating the microbiota on the phenotype.

At Quinten, the culture of subgroup analysis is present because to meet the need for transparency of results so that decision-making by domain experts remains possible and efficient. In this way, an algorithm called *Q-Finder*[4] [Esn+20] has been developed to generate statistically credible subgroups to answer clinical questions in particular. We used this algorithm as a basis for our research to improve it and develop a more appropriate approach to metagenomics.

1.2.3.3. *Q-Finder*

Two main parts of SD algorithms are defined as: *Rule refinements*, the rules exploration phase, and *rule selection*, the step keeping or removing rules. In many SD algorithms both parts have the same heuristics but more recent approaches have used different ones [SJF14; Val+17b]. *Q-Finder* is in the case where different heuristics are driving the results, it combines an exhaustive search (*rule refinements*), with a cascade of filters (*rule selection*) based on metrics assessing key credibility criteria, including relative risk reduction assessment, adjustment on confounding factors, individual feature's contribution to the subgroup's effect, interaction tests and adjustment tests (multiple testing). This allows *Q-Finder* to directly target and assess subgroups on recommended credibility criteria.
A preliminary work was to place and benchmark *Q-Finder* with other algorithms in rules learning literature. As *Q-Finder* is constructed to perform both SI and KDD analysis, it has been compared accordingly to other algorithms on the database of the International Diabetes Management Practice Study (IDMPS). The goal was to better understand the drivers of improved glycemic control and rate of episodes of hypoglycemia in type 2 diabetics patients. We compared *Q-Finder* with state-of-the-art approaches APRIORI-SD [KL07] and CN2-SD [Lav+04] for KDD algorithms, and Virtual-Twins [FTR11] and SIDES [LD14] for SI algorithms. The results demonstrate its ability to identify and support a short list of highly credible and diverse data-driven subgroups for both KDD and SI tasks.

[4]Proprietary algorithm of Quinten company

1.2.3.4. Adapting subgroup discovery to interpretable metagenomic model for dysbiosis classification

Recent studies have shown that altering the microbiome is a strong way to cure metabolic disease and recover a healthy profile. Indeed, treatments are going to rid of immune suppression and drug therapy to promote manipulation of the intestinal flora instead. These treatments could refer to dietary changes, specific microbial manipulation and fecal microbiota transplantation [Sco+15]. There are many variations between the microbiomes of individuals that need to be better understood [Ain20]. For these reasons, we have taken the direction in this thesis to address the problem of interpretable prediction of metagenomic data.

By constructing a signature as a set of combinations of sub-signatures characterizing different subgroups of patients, the technique allows a double classification: one of the pathologies and one of the subgroups to which a patient belongs. Using a credible subgroup discovery algorithm, allows an identification of different classification sub-signatures, each classifying subgroups of individuals and each using a different subset of metagenomic taxa. This makes it possible to account for the known multiple natures of the causes of metagenomic dysbiosis and to offer a personalized explanation. Consequently, it prospects for targeted treatments while maintaining classification performance at the level of the state of the art.

Method: The algorithm we proposed, named *Q-Classifier*, is a combination of subgroup discovery and supervised classifications approaches. It is therefore an interpretable method which is distinguished by three key points:

- The signature is interpretable and personalized. Indeed, the signature (i.e. the description of the metagenomic signature) is not represented as a sub-signature, a regression formula, or a decision tree. Instead the signature is represented by a set of rules (sub-signatures) for different groups of individuals. This approach allows an explanation of the classification decision that is not only interpretable but also "personalized" because it is different for each group of individuals. A patient can correspond to all or part of the sub-signatures.
- A different statistical measure of credibility for each signature rule is provided indicating a level of confidence in its applicability (e.g. Odd-ratio or F1-score and p-value).
- When an individual cannot be correctly classified, it is then rejected and delegated to be classified by a state-of-the-art algorithm.

Then, these characteristics describe the algorithm as a reject and cascade classifier. Cascading is a multi-step system in which only observations not classified by the first classifier are delegated to a second classifier [HB16a]. The section 4.3 of chapter 4 details how the algorithm work.

Results: The *Q-Classifier* proves that it is capable of achieving state-of-the-art approaches on real-world datasets, but when the data is preprocessed with a CLR transformation, it achieves even higher results. The percentage of samples rejected by the subgroup discovery step depends on the database and ranges from 20% to 77%. It is often related to the prediction scores, the higher the scores, the lower the rejection rate. We were also able to verify with the simulated datasets that the algorithm was able to recover the artificially created metagenomic abundance profiles. The *Q-Classifier* has also been trained on the abundance tables output from *Metagenome2Vec* and obtained promising results slightly below the ones obtained on the abundance tables calculated by the *MetaPhlAn2* algorithm from the Pasolli et al. [Pas+16] study.

1.2.4 Scientific mediation

Accepted papers

- Cyril Esnault, May-Line Gadonna, Maxence Queyrel, Alexandre Templier, and Jean-Daniel Zucker. “Q-Finder: An Algorithm for Credible Subgroup Discovery in Clinical Data Analysis — An Application to the International Diabetes Management Practice Study”. *Frontiers in Artificial Intelligence*, Dec 2020. doi: 10.3389/frai.2020.559927
- Maxence Queyrel, Edi Prifti, Alexandre Templier, and Jean-Daniel Zucker. “Towards End-To-End Disease Prediction From Raw Metagenomic Data”. *International Journal of Biomedical and Biological Engineering*, June 2021. url: https://publications.waset.org/pdf/10012114
- Maxence Queyrel, Alexandre Templier, and Jean-Daniel Zucker. “Reject and cascade classifier with subgroup discovery for interpretable metagenomic signatures” *Advances in Interpretable Machine Learning and Artificial Intelligence*, July 2021. Publication in progress.

Patent

- Maxence Queyrel, Alexandre Templier, and Jean-Daniel Zucker. “Patient classification method from sets of pathology-specific sub signatures.”

2 Experimental methods and design

Our thesis work led to the creation of algorithms seen in the previous chapter (Section 1.2): end-to-end processing with deep learning (chapter 3) and interpretable prediction with subgroup discovery (chapter 4). To carry out the study of the different approaches, we have determined several experimental settings. All methods were tested on several datasets with different software. This chapter summarizes the characteristics of the datasets and tools used during the thesis. First, real-world metagenomic dataset are described, then the strategies and objectives of the metagenomic simulations are specified, next the IDMPS database is introduced and finally frameworks and technologies used to develop and execute the algorithms are detailed.

2.1 Survey of existing metagenomics datasets

Existing metagenomic datasets generated by Illumina shotgun sequencers are used in several studies [EDF15; Tho+19; Pas+16; Pri+20]. These datasets represent a basis for comparing our different methods with the state of the art. They are accessible from NCBI website[1] which repertories open-source studies. The datasets span four diseases associated with gut microbiota metagenomic data: colorectal cancer [Zel+14], liver cirrhosis [Qin+14], obesity [Met+13] and type 2 diabetes [Qin12]. The table below summarized the four main public datasets manipulated in this thesis:

Dataset name	Disease	Control subjects	Case subjects	Control-to-Case	Size[2]	Reference
Colorectal	Colorectal Cancer	73[3]	48	60.3%	$\sim 480Go$	[Zel+14]
Cirrhosis	Liver Cirrhosis	114	118	49.1%	$\sim 1.1To$	[Qin+14]
Obesity	Obesity	89	164	35.2%	$\sim 1.3To$	[Met+13]
T2D	Type 2 diabetes	174	170	50.6%	$\sim 1To$	[Qin12]

Tab. 2.1.: Information about the four real-world metagenomic datasets

[1] www.ncbi.nlm.nih.gov

Dataset name	Disease	Control subjects	Case subjects	Control-to-Case	Size
Colorectal	Colorectal Cancer	73	48	60.3%	$\sim 480Go$
Cirrhosis	Liver Cirrhosis	114	118	49.1%	$\sim 1.1To$
Obesity	Obesity	89	164	35.2%	$\sim 1.3To$
T2D	Type 2 diabetes	174	170	50.6%	$\sim 1To$

Tab. 2.2.: Information about the four real-world metagenomic datasets

In order to combine the main research directions of the thesis and to facilitate their comparison, these datasets are used in the experiments related to both deep learning and subgroup discovery algorithms. The datasets' raw data, i.e the reads obtained by NGS technologies, are the input of the deep learning workflow *Metagenome2Vec* (chapter 3), whereas the input of subgroup the discovery algorithm *Q-Classifier* (chapter 4) is the species abundance tables obtained with the *MetaPhlAn2* algorithm [Tru+15] from the work of Pasolli et al. [Pas+16].

2.2 Simulating metagenomic datasets

The simulation of metagenomic data makes it possible to get rid of sequencing machines (leading in a gain of time, cost, etc...) in order to create artificial samples. It has several advantages such as:

- Defining the number of samples and their size.
- Determining all the genomes present in the ecosystem.
- Assigning the class, i.e. the genome, of each generated read.
- defining the prevalence of genomes per sample.

In our experiments, the metagenomic data simulations are used in two ways:

- To generate a set of reads to learn embeddings or train a classifier to a certain taxonomic level
- To generate metagenomic samples from specifically defined abundances profiles, resulting in a dataset with artificial control and case samples to train and evaluate our models on a prediction task.

[2] Datasets have a massive size considering they only have few hundreds of patients. Metagenomes are composed of ~ 80 million reads, each one composed of ~ 90 nucleotides.

[3] 15 patients had an adenoma; this is a benign tumor, so they have been labeled as control cases as in Pasolli et al. [Pas+16].

The simulation process needs reference genomes to work as it takes random DNA fragments from genome sequences. To select these genomes, DNA sequences from a catalog of 9,879,896 genes [Met+14a] have been projected onto reference genomes database available on NCBI. Then, the most abundant genomes have been retained to compose a set of 506 different genomes representing 235 species, 79 genera and 37 families. These genomes are given as input of the *CAMISIM* software [Fri+19] to simulate Illumina reads or Nanopore reads when it is combined with the *NanoSim* software [Yan+17].

2.2.1 Datasets used to train embeddings and taxa classifier

The simulated dataset to train the embeddings and the taxon classifier must consist of a supervised metagenomic read dataset that the model will learn to classify. To deal with the problem of unbalanced classes leading the model to learn more of the majority classes, we need to ensure that the simulated dataset has a uniform distribution of taxa. This parameterization can differ from one software to another. Indeed, *CAMISIM* and *NanoSim* take a file of genome abundances as configuration but manage it differently. *CAMISIM* takes as input the size of the genome in addition to its abundance. Thus, to generate an abundance equally proportionate between genomes we must use the following formulas to calculate the abundance given as input to *CAMISIM*:

$$\begin{aligned} A_g &= \frac{1}{L_g} \\ AG_g &= \frac{A_g}{\sum_i^{|G|} A_i}, g \in G \end{aligned} \tag{2.1}$$

Where G is the set of genomes, L_g is the base pair length of the genome g and AG_g stands for the equally balanced abundance between genomes for the genome g. For *NanoSim*, AG_g is just defined as a uniform distribution from the number of genomes like $AG_g = \frac{1}{|G|}$.

We have outlined the case where the classifier is trained to the strain level. However, if the model is trained to another taxonomic level, such as species, the formula to calculate the abundance of species should be modified accordingly. Indeed, if genomes come from the same species, the simulation will produce more reads for

that species than one where only one genome comes from it. The formulas used to avoid a species appears too often or not often enough are defined by:

$$
\begin{aligned}
L_s = \sum_{i}^{|G_s|} L_i \ ; \ L\_norm_s = \frac{L_s}{\sum_i^{|S|} L_i}, \ s \in S \\
A_g = \frac{1}{L\_norm_s \times (L_g / L_s)} \ ; \ A_g = \frac{A_g}{\sum_i^{|G_s|} A_i} \\
AS_g = \frac{A_g}{\sum_i^{|G|} A_i}
\end{aligned}
\tag{2.2}
$$

Where S is the ensemble of species, G_s is the ensemble of genomes for the species s, L_s is the base pair length of the species s and AS_g stands for the abundance equally balanced between genomes at the species level for the genome g.
3.5M and 1.5M Illumina reads with an average of 150 base pairs have been simulated for the train and validation set respectively, corresponding to a depth of coverage (mapping depth) of 27% over all the genomes. For Nanopore data, a total of 735K training data and 315K validation data with a base pair average of 2590 have been generated, corresponding to a mapping depth of 100% over all the genomes. The initial given abundance was calculated with the formulas 2.2 to be equally proportionate at the species level. Indeed, the model is trained at the species level rather than the genome level because genomes of the same species are close enough to have the same prediction and it is easier to train the model with a smaller number of classes. An almost equal number of sequences for each species is not representative of real metagenomes where abundance follows exponential distributions. Nevertheless, in the case of read classification modeling, this prevents the classifier from focusing and predicting the predominant classes while learning a more robust embedding. These simulated data have the advantage over real NGS data of providing information on the origin of a sequence, allowing the training of a supervised algorithm and test its performance in estimating abundance profiles resulting from the classification of reads on taxa.

2.2.2 Datasets to learn the disease prediction tasks

The simulated metagenomic dataset, used to train the sample group stratification models, must define several taxonomic abundance profiles characterizing the different individuals. Thus, these profiles must be close and distant within the same group to allow the algorithm to classify them together without it being too trivial. Artificially created abundance tables allow us to simulate metagenomic reads for each sample to evaluate the algorithms against known profiles [FA12].

Different strategies to simulated abundance profiles: Some studies are working on discovering interaction between pairs of microbial taxa with metabolic network models [FA12; LB13] or with Lotka-Volterra model [SPW16]. These methods provide information on the correlation of taxa allowing a better control on the credibility of the metagenomic profiles created. Weiss et al. [Wei+16] have detailed several methods for metagenomes simulation such as *Null model*, *Ecological*, *Lotka-Volterra* and *Time Series*. *Lotka-Volterra* models are a system of differential equations and *Time Series* is based on time series equation between taxa. The *null model* creates abundance from an initial null distribution with different strategies like drawing samples with Dirichlet distribution without sum-constrained (raw abundance) or with sum-constrained (relative abundance) and reducing or increasing species diversity. Finally, the *Ecological* method is based on linear relationships between taxa abundance where, mutually or independently, abundance may increase or decrease with the absence or presence of certain taxa. For our experiments, the simulations have been done using both *null model* and *ecological* strategies.

2.2.2.1. Null model

This approach is used to generate two datasets, one with Illumina generator named *Null Model Illumina*, and the other with the Nanopore generator named *Null Model Nanopore*.

The *Null Model Illumina* dataset is created to allow the learning of the end-to-end deep learning model and to test its efficiency to find the bacterial species that have an impact on the classes of the samples. To create this dataset, an artificial disease is simulated based on known bacterial genomes and abundance profiles which significantly vary in abundance between two groups (namely control vs case). The control case samples are simulated with the same abundance of species, so the number of reads is almost the same for each species. In contrast, the samples from case patients have an abundance arbitrarily set to 3 times more important for five species that imply the disease. In total, 200 profiles were simulated with an average number of reads per sample of about 164k representing a total of 10GB.

The *Null Model Nanopore* dataset is created to evaluate *Metagenome2Vec* and *Q-Classifier* models on Nanopore reads from known species abundances. We choose to draw the initial abundance distribution from the cirrhosis data set [Qin+14]. In this data, some variations in metagenomic abundances between individuals are known in the literature to cause cirrhosis. It allows us to more easily evaluate the models training in terms of classification and explicability. For each sample in the dataset, 3

artificial samples are created by adding Gaussian noise to the original abundance of each species present. This generates a dataset 3 times larger than initially to train the models, while limiting the possible similarity between samples created from the same profile. Due to the compositional nature of the relative abundance, the added Gaussian noise is in accordance with equation 2.3 to meet the data constraints: retrieve values between $[0., 1.]$ with a sum equal to 1 for each sample.

$$\begin{aligned} x_i = logit(a_i) = log(\frac{a_i}{1-a_i}), a_i \in [0., 1.] \\ x'_i = x_i + \epsilon_i \\ a'_i = sigmoid(x'_i) = \frac{1}{1+exp(-x'_i)} \end{aligned} \tag{2.3}$$

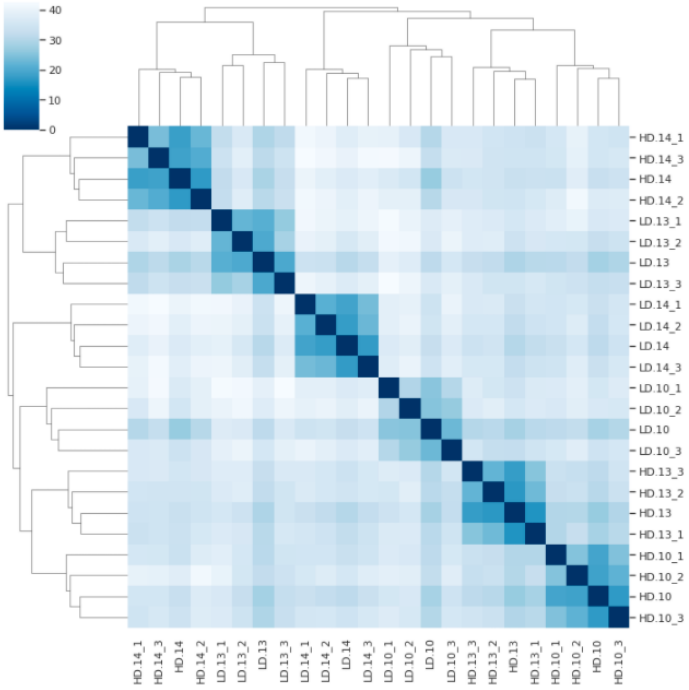
Where a_i is the initial abundance and a'_i is the result with the added Gaussian noise. The Logit function is used to unconstrain the abundance and the Sigmoid function is used to retrieve the original constraint. We must control the added noise so that it is:

- Not too low: resulting in an overfitting because the simulated data are too close to the initial data
- Not too High: producing simulated data too far from reality, preventing the models from successfully learning a correct stratification.

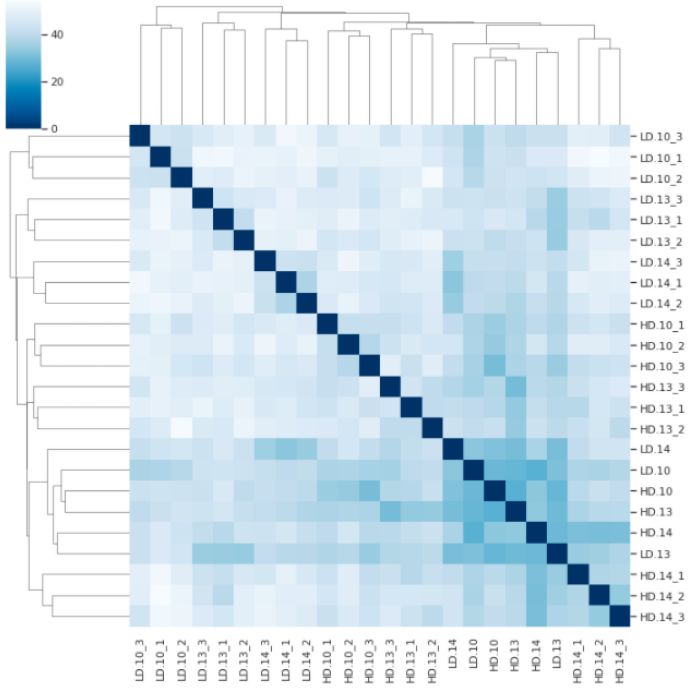
To check these characteristics, the distance between the abundance of each profile is calculated to analyze how close or far apart they are (Figure 2.1). The Aitchison distance is used because it is better suited for compositional data and is defined by:

$$A = E(clr(x), clr(y))$$

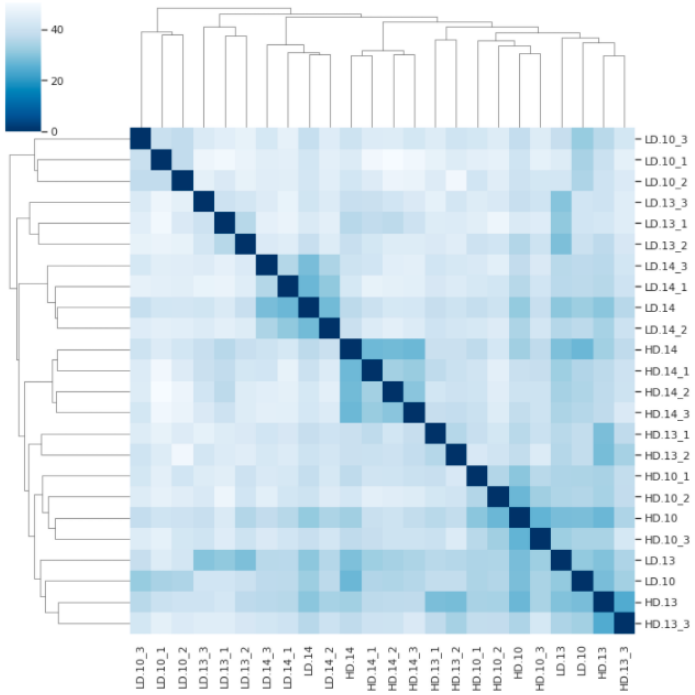
Where A is the Aitchison distance, E the Euclidean distance ($E = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$), x and y are two vectors of relative abundance and clr is the centered log-ratio transformation explained in section 1.1.6.



(a) Noise = 0.5



(b) Noise = 5.0

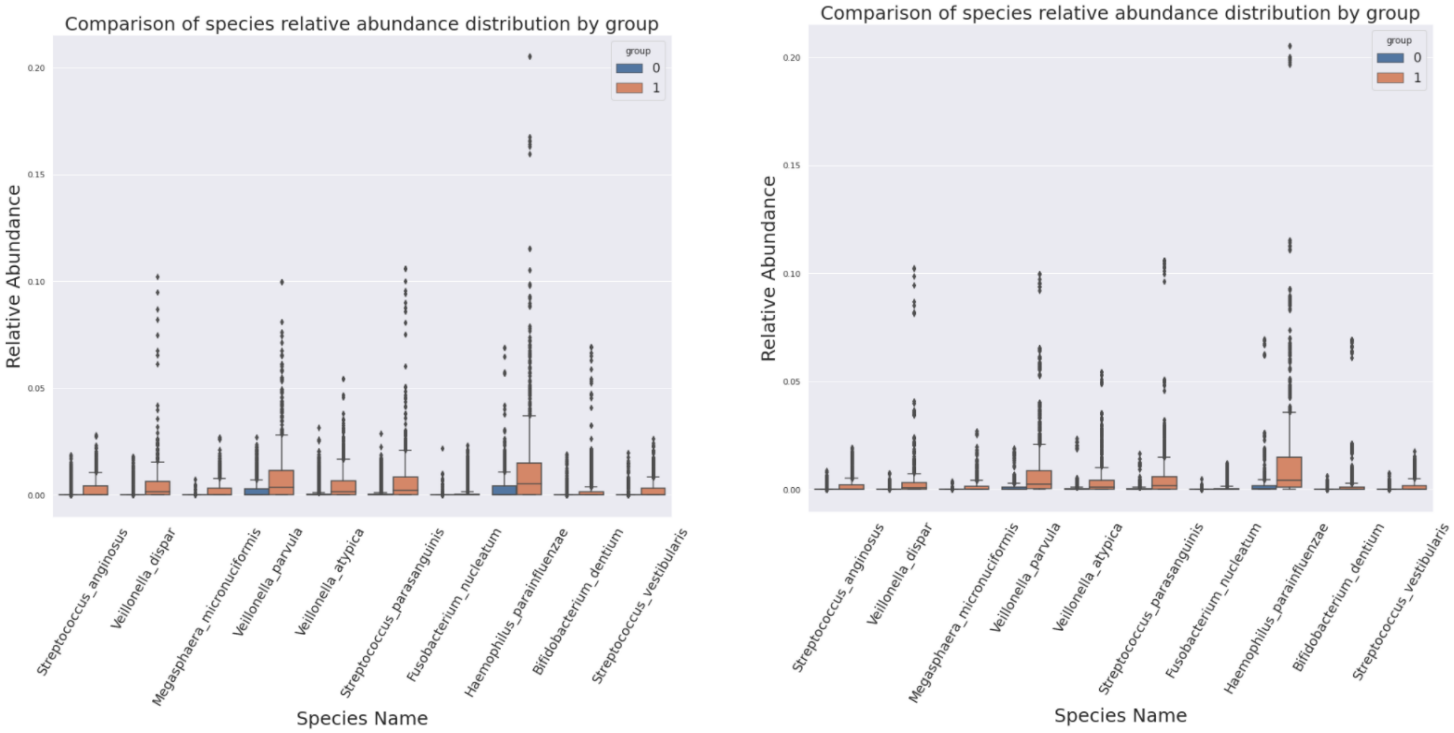


(c) Noise = 2.5

Fig. 2.1.: Cluster map of the Aitchison distance between the relative abundances of 3 initial control profiles (HD), 3 initial case profiles (LD) and their 3 noisy simulated samples. The underscore followed by a number is used to designate a simulated profile.

The figure shows that the noise is too low when it is equal to 0.5 because the clusters between a profile and those generated from it have too close Aitchison distances. When the noise is equal to 5., it is too high because a cluster appears between the initial profiles, whether they are control or case, implying that even with respect to the class, the simulated profiles are even more distant. Considering these details, a $noise = 2.5$ seems to be a good compromise to balance the Gaussian noise applied on the initial profiles.

It is also important to check whether the abundance of species that differ most from the two classes, control and cirrhosis, in the original data set are still different when simulated samples were added (see Figure 2.2).



(a) Initial dataset

(b) Dataset of simulated profiles added with a noise equal to 2.5

Fig. 2.2.: Box plot of the species abundance distribution for the control and cirrhosis group. The ten species plotted have obtained the highest value on the Mann-Whitney test meaning that the distributions of the two groups are the most distant.

The distribution of abundance in the two datasets is quite similar, which implies that noise has not deterred the initial assumptions about species. After the different checks, the noise is set to 2.5 and considering the 232 initial examples, the total of profiles in the final database is equal to 928. The average number of reads by sample is around 28k and represent a total of 29Go.

In addition to the *Null Model* strategy for metagenomic simulation, the *Ecological* process has also been tested in our experiments.

2.2.2.2. Ecological

The *Ecological* simulation process has a holistic view that allows us to generate profiles and to know interactions between species (*Ecological Nanopore* dataset). It is suitable for evaluating models on biomarker discovery in addition to stratification. The simulation strategy here is to randomly take a single control sample from the cirrhosis dataset [Qin+14] and change its abundance 500 times to create 500 control (resp. case) samples. We do not know, from the literature, a number of relevant profiles for this type of simulation. That is why, we arbitrarily chose to create two control (resp. case) profiles, in which there are two species associated with an altering factor of abundance characterizing each new profile. The altering factors are defined in such a way that the abundance of other species is not or not significantly modified. This is a way of preserving the initial value of abundance because, as this is compositional data, changing some values will also alter others. This is preferable to be avoided for biomarker discovery assessment. Each sample randomly takes 1 or 2 profiles with a probability of 0.5, the abundance of the species included in the profiles is then modified with a predefined factor, finally a Gaussian noise is applied as in the *null model* strategy to be sure that the profiles are all different. There are 1000 created profiles with an average number of reads equal to 28k by sample representing a total of 31Go.

The characteristics of the 3 simulated datasets used for disease prediction tasks (sections 3.4 and 4.3.4.2) are summarized in the following table:

Dataset name	Simulation method	Control subjects	Case subjects	Control-to-Case	Size
Null Model Illumina	Null model	100	100	50.0%	$\sim 10Go$
Null Model Nanopore	Null model	456	492	48.1%	$\sim 29Go$
Ecological Nanopore	Ecological	500	500	50.0%	$\sim 31Go$

Tab. 2.3.: Information about the three simulated metagenomic datasets

We have detailed the different real and simulated metagenomic datasets used during the thesis. However, we manipulate another dataset, called IDMPS, a clinical study on diabetes. It is useful in our research for the subgroup discovery part. Its description and use in our experiments are detailed in the next section.

2.3 Introduction of the IDMPS database

The International Diabetes Management Practice Study (IDMPS) database is used during the experiments to benchmark the Q-Finder algorithms against state-of-the-art subgroup discovery algorithms. The dataset is not related to metagenomic data, so its task is mainly focused to evaluate the results of the interpretable models in chapter 4.

IDMPS is an ongoing international, observational registry conducted in waves across multiple international centers in developing countries since 2005. Each wave consists of a yearly 2-weeks fact-finding survey, which aims to document in a standardized manner: practice environments, care processes, habits, lifestyle and disease control of patients with diabetes under real world conditions. It has recently led to new findings related to the sub optimal glycemic control in individuals with type 2 diabetes in developing countries and the need to improve organization of care [Asc+20]. Observational registries for patients suffering such conditions are pivotal in understanding disease management. In 2017, an estimated 425 million people were afflicted by diabetes worldwide, with Type 2 Diabetes Mellitus (T2DM) accounting for approximately 90% of cases. By 2030, diabetics should represent 7.7% of the adult population, or 439 million people; and 629 million people by 2045 [CMZ12; SSZ10; Ogu+17]. The two most recent waves to date of IDMPS (wave 6 [2013-2014] and wave 7 [2016-2017]) were selected for the following experiments, including data from 24 countries from Africa, Middle East, India, Pakistan, Russia and Ukraine. Only data from patients having T2DM and taking either a Basal insulin, a combination of Basal and Prandial insulin or a Premixed insulin were included.

2.4 Code implementation

2.4.1 State-of-the-art classifiers

The Scikit-Learn 0.23.2 package is used to train state-of-the-art classifiers such as *Support Vector Machine (SVM)*, *Random Forest (RF)*, *AdaBoost (AB)* or *Gradient Boosting (GB)*. Parameters search algorithms to optimize models comes also for this package.

2.4.2 End-to-end deep learning for disease classification from metagenomic data

The full pipeline of the end-to-end deep learning disease classification is developed to preprocess DNA reads from fastq files and to train k-mers embeddings, reads embeddings, classifier of reads into taxa and classifier of metagenomes into disease. The source code is available in a Git directory at the following link `https://github.com/MaxenceQueyrel/metagenome2vec`. The algorithms and frameworks used for experiments are summarized below:

Data preprocessing Given the amount of data we decided to take advantage of the Spark 3.0.2 Python Framework [Zah+12] running on distributed clusters to manage memory and make parallel computing on CPU for all data processing. We use YARN (UMMISCO cluster), Torque (UMMISCO and MeSu cluster) or SLURM (Julio-Curie cluster) as cluster managers to change the resources allocated for each of our experiments. The deep learning models are always trained on GPU computing.

Machine learning We use PySpark 3.0.2 to train the *word2vec* algorithm. *GloVe* and *FastText* algorithms are directly downloaded from GitHub, written in C and C++ respectively. For state-of-the-art classifier models, refer to section 2.4.1. Finally, the neural network architectures are implemented with PyTorch 1.7.1 [Pas+19] using GPU computation. We detail the packages and algorithms used for each sub part of the *metagenome2vec* workflow:

- *kmer2vec*:
  - *word2vec*, from PySpark 3.0.2.
  - *GloVe*, version 1.2 from GitHub written in C.
  - *FastText*, version 9.2 from GitHub written in C++.
- *read2vec*:
  - *FastText* mean embeddings aggregation.
  - Language model transformers, trained with Pytorch 1.7.1.
  - FastDNA, from GitHub written in C++.
- *read2genome*:

  - Multiple Layers Perceptron on top of read embeddings, H2O sparkling water 3.32.
  - FastDNA, from GitHub written in C++.
- *metagenome2vec*:
  - *SVM*, *RF*, *GB* and *AB* (section 2.4.1) for classification benchmark on *metagenome2vec* vectorial representation and Bag of K-mers (BoK) representation.
  - DeepSets and *Variational Auto Encoder (VAE)* models are design with Pytorch 1.7.1. Model's hyper parameters optimization is computed with Ax 0.1.20 package for Bandit and Bayesian search algorithms combined with ray-tune 1.3.0 package for resources allocation and parallelization.

2.4.3 Generate statistically credible subgroups for interpretable metagenomic signature

- The *Q-Finder* algorithm is designed in C++ with an API called by a python package. The whole workflow is built on Dataiku V8 by decomposing each part of the algorithm into different Python 3 modules in order to have a better flexibility of implementation (to filter the rules for example). The code is parallelized on a local cluster with Mesos as cluster manager.
- The *Q-Classifier* algorithm uses the *Q-Finder*'s rules generation and follows the same logic of Python modules implemented on Dataiku. The state-of-the-art classifiers trained in this part are *SVM*, *RF*, *GB* and *AB* (section 2.4.1)

2.5 Conclusion

As described, several datasets with different schemes are used to learn and evaluate our models. Moreover, the numerous packages, technologies and heterogeneous computational resources led to an important work of code adaptability in order to aggregate all these elements between them. The implementation of the algorithms is obviously an essential point of this thesis but handling the large volume of data to be adapted according to the clusters and algorithms was often a more time consuming and complex aspect although less at the center of our research.

3 End-to-end deep learning for disease classification from metagenomic data

One critical task, when using microbiome data in a precision medicine context, is to discriminate diseased patients from healthy or within different severity groups [Job18]. In this thesis, we want to solve this problem without depending on external resources (e.g. gene catalogs) during the prediction phase and we would like our method to be both faster and at least as efficient as the state-of-the-art. Deep learning is best suited to create useful and reusable representations on complex data such as images, texts, time series, etc, as such it is potentially a good approach to deal with metagenomic data. We were interested in various methods to successfully project metagenomes into a latent space characteristic with a discriminative power. This chapter details how metagenomic data are represented mathematically and how they relate to natural language processing (NLP). It also highlights the state-of-the-art models related to the creation of embeddings in metagenomic. Finally, the methods proposed in this thesis are explained and detailed through different experiments and results.

3.1 The representation of metagenomic data

Representing Metagenomes To mathematically represent the different concepts of our approach, let $D = (x_m, y_m)_{m=1}^{M} = \{X, Y\}$ denote a set of M metagenomes and the associated labels $Y \in \{0, 1\}^M$. A metagenome x_m is composed of $R_m \gg 10^6$ DNA reads. A DNA read $s_r, r \in \{0..R_m\}$ is a sequence of length $L_r \in 50 \sim 200$ for Illumina or $L_r \in 10^4 \sim 10^6$ for Nanopore technology. The reads are formed by several nucleotides in the vocabulary $\mathcal{A} = \{A, T, C, G\}$, so $s_r \in \mathcal{A}^{L_r}$.

Representing reads In NLP, there are explicit word and sentence delimiters. On the contrary, in the case of DNA reads there is no explicit information to systematically delimit sub-sequences. Moreover, it is difficult to know the location of a read in

genomes because the DNA were fragmented prior to sequencing and the location of the reads is lost after sequencing. To transform the reads onto something similar to words, a possible approach may be to simply split the sequences into k-mers [MV19; Wol+18; Min+17; Lia+20]. Various size of k can be considered depending on the task. Padding between k-mers is equal to one because k-mers refers to all sub-sequences of length k. For example, if we have a sequence of seven bases "AATCCGA" and if $k = 3$, then after splitting we do not only obtain the k-mers "AAT" and "CCG" but also "ATC", "TCC" and "CGA".

Building k-mers, reads and metagenome embeddings Similar to the NLP applications, using vector representations of k-mers can overcome some limitations to treat reads as discrete atomic sequences of nucleotides. Similarly, vector representation of reads and metagenomes can be envisioned to go beyond their simple encoding representations [Wol+18]. In our work, we focus on learning metagenome embeddings that could both reduce the data dimensions and also support computationally efficient predictive models from embedded raw metagenomes. As Metagenomes are composed of reads and reads are composed of k-mers, it is natural to consider a multilevel embeddings approach. This is the reason why, in section 3.3, we introduce and detail three main stages of data transformation: *Kmer2Vec*, *Read2Vec* and *Metagenome2Vec* to compute respectively vector representations of k-mers, reads and metagenomes.

3.2 State of the art

Several recent DL methods have been adapted to support the analysis of genomic or metagenomic data for different tasks. The differences between these studies are the representation of DNA sequences, the types of algorithms and obviously the final objective. The Figure 3.1 summarizes the different metagenomic embeddings models in relation to our approach *Metagenome2Vec*. There is already a large literature related to DL approaches based on examples represented as DNA sequences in the context of genomics or metagenomics studies. For information, genomic data focus on the genetic material of an individual, whereas metagenomic data concern several individuals in the same environment. As a result, the sequence alignment process is relatively easier on genomic data than on metagenomic data. That's why, these DL methods typically operate directly on raw metagenomic sequences and operate on aligned genomics sequences.

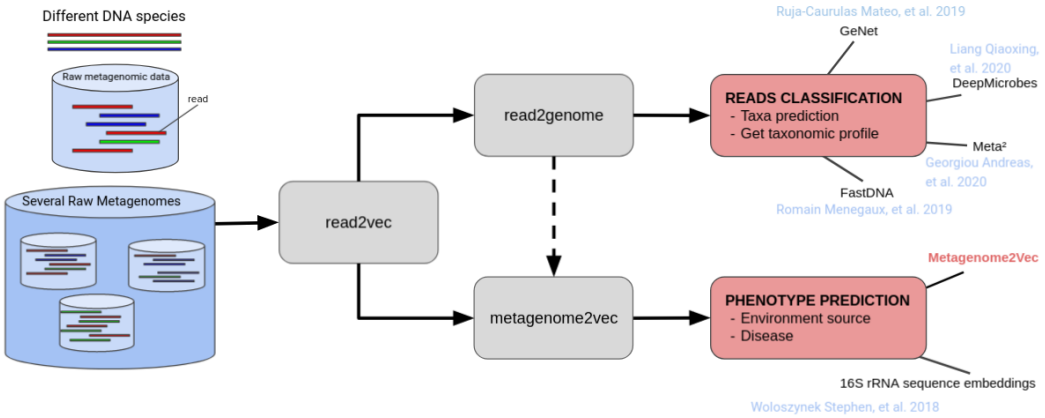


Fig. 3.1.: Workflow of metagenomic data projected into low-dimensional representation with embedding learning algorithms along with SOTA approaches. The blue color represents the input data, the grey color represents different internal modules of the pipeline and the red color the prediction task performed. The dotted line is only a part of the *Metagenome2Vec* algorithm. Algorithms written (including *Metagenome2Vec*) are linked with their corresponding task. We can see that *Read2Vec* is a module for both phenotype and read classification. If the abstraction is at the read level, results are handled to classify reads for taxonomic profiling. If the abstraction is at the metagenome level, prediction could be used for phenotype classification

3.2.1 Machine learning models from nucleotide one hot encoding

It is well-known that recurrent neural networks (RNN) as well as convolutional neural networks (CNN) both work well on sequential data [CCC16; You+18]. That is why, Genomic studies have used both architectures on DNA sequences. One classical genomic task is to analyze *chromatin accessibility*. It involves finding some regions of the genome that are accessible for cellular machines involved in gene expression while others are compactly wrapped and remain inaccessible. Methods exist to measure the accessibility of chromatin related to gene expression, but it is expensive and time consuming. Different works have shown that RNN or CNN on DNA sequences can capture relevant patterns and outperform state-of-the-art methods in chromatin accessibility prediction for instance (*DanQ* - [QX16], *DeepSEA* - [ZT15] or *Basset* - [KSR16]). A benchmark of these approaches was introduced by Hildick-Smith and Bahtchevanov [HB16b].

In metagenomics, some studies focus on the hierarchical taxonomy structure and are facing a multi-class classification problem. This is the case of *GeNet* [Roj+19], a DL model based on CNN and ResNet architecture [He+15]. Authors have used a one-hot encoding of the input nucleotides and their position in the read. The

loss function is computed at each taxonomy level and the prediction at any level is forwarded to the next one adjusting the decision of the classifier.
However, all these algorithms keep the initial representation of DNA and simply one-hot encode the four nucleotide bases {A, C, G, T}. In other words, most algorithms operate on a $4 \times L_r$ matrix where L_r is the sequence length. This representation is quite basic and does not consider the similarities between k-mers. This is comparable to representing a sentence as a set of letters rather than a set of words.

3.2.2 Machine learning models from DNA embeddings

An emerging idea have been to no longer work only with single nucleotides but with k-mers. Considering such subsequences, it makes it possible to train models that create vector representations and capture relationships between each k-mers. Research in NLP has seen a major development on low-dimensional representation of words. These methods regularly outperform the simple version of bag of words by projecting words into a vector representation that accurately captures syntactic and semantic word relationships. Recently, based on this concept, there have been some approaches considering k-mers embeddings. In the work of Ng [Ng17], k-mer embeddings are computed with a *word2vec* algorithm. A relationship is highlighted between the arithmetic operation on *word2vec* vectors and the corresponding concatenation of k-mers. Similarly, in the Min et al. [Min+17] study, where the goal is to classify chromatin accessibility (as explained in the previous section), the *GloVe* [PSM14] is used to create k-mer embeddings before training the final neural network. Experiments have shown that results are better when the sequence is transformed into k-mer embeddings. Nevertheless, Min et al. [Min+17] set the k-mer size to 6 without discussing other configurations that could have demonstrated the overall importance of this parameter.

There have been also several attempts to learn ML models directly from raw metagenomic data (see Figure 3.1). Most of them address the task of predicting the origin of reads (called taxonomic profiling) at a certain taxonomy level or to perform phenotype classification. Metagenomic data is a set composed of millions of reads, so it requires transformations into a suitable representation before training prediction models. It is possible to get a lower representation of metagenomes at the k-mer / sequence level for taxonomic prediction and at the sample (metagenome) level for phenotype prediction. To assign taxonomic information to each read, the authors of *FastDNA* algorithm [MV19] have demonstrated that their approach using embeddings of k-mers achieves performances comparable to the state-of-the-art. In the first step of their approach, they define the length k of the k-mers that describe

the DNA sequences. Then they run the *FastText* algorithm [Jou+16] to learn low-dimensional embeddings (dimension from 10 up to 1000). All k-mers in a sequence are replaced by their corresponding vector and summed to get an embedding of the read they belong to. Then this new vector is passed to a linear classifier, which computes the softmax function and minimizes the cross-entropy loss by gradient descent. The k-mer embeddings are directly learned from the read classification considering the result of the loss function. The authors demonstrated that significant prediction results appear with a k-mer size greater than 9 nucleotides, especially for values equal to 13 or 14. With a similar objective, Liang et al. [Lia+20] propose *DeepMicrobes*, a neural network with an architecture composed of an embedding of k-mers, a bidirectional LSTM, and a self-attention layer to predict the species or the genus of a read. In their experiments, k-mers of size k=12, lead to their best results.

In the work of Woloszynek et al. [Wol+18], the objective is to add, in addition to taxonomic profiling, a method to retrieve the source environment of a metagenome (phenotype prediction). A *Skip-gram word2vec* algorithm [Mik+13] is trained for k-mers embeddings and a SIF algorithm [ALM17] is used to create reads and samples embeddings. They demonstrate the usefulness of such an approach for clustering and classification tasks. Moreover, they show that such embeddings allow to train models using larger k-mers (k greater than 9), which is not possible when relying on simpler representation such as one-hot encoding because their size grows exponentially grows with k.

We have seen that k-mers play an important role in the transformation of metagenomic data into embeddings. They are the representation in a latent space of the k-mers dictionary and can be used to build read or metagenome embeddings. Since metagenomic data are composed of multiple reads, other studies have sought to manipulate these data using Multiple Instance Learning (MIL) models [WZ00] where a class label is assigned to a bag of instances.

3.2.3 Learning from multiple-instance representation of reads

A quite different approach named *RegMIL* [RR18] uses MIL in order to assign disease to each sample. Unlike methods described above, *RegMIL* does not train a model on raw data but first begins by performing sequence assembly, binning and clustering contigs. K-mers are counted and normalized for each sequence in clusters. A neural network regression model is then trained to compute a score corresponding to the association between k-mers and the disease. Finally, a random

forest classifier is trained on top of this representation to compute phenotype prediction. $META^2$ [Geo+20] is another algorithm using MIL. The model, based on *GeNet* or *DeepMicrobes* with an additional MIL pooling layer, is trained to predict the distribution of the reads set for each taxon. The MIL layer increases the model's prediction scores and the authors suggest that this improvement is due to the model's exploitation of the species co-occurrence matrix.

Our *Metagenome2Vec* method is mainly oriented towards the aspects of creation of embeddings from raw metagenomic data as well as on the learning of MIL model.

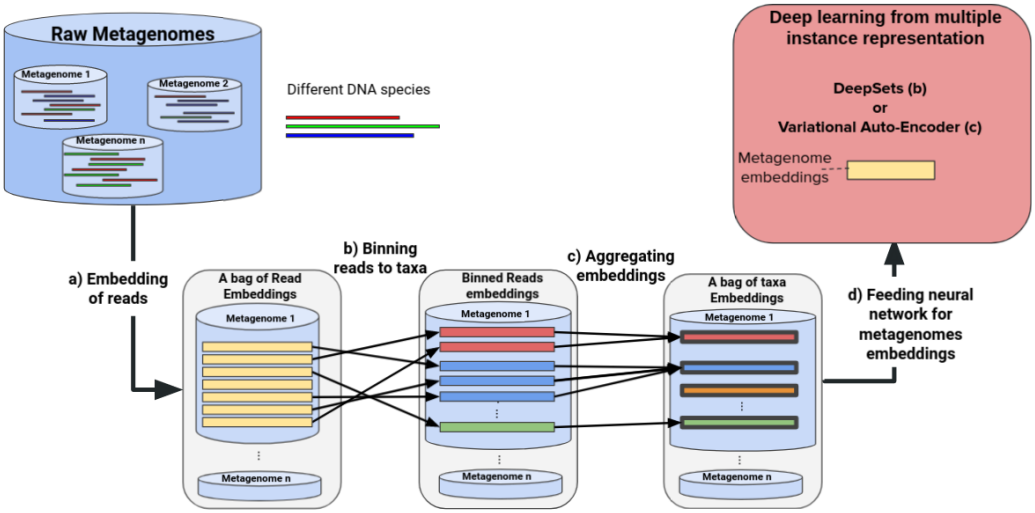
3.3 *Metagenome2Vec*: a novel approach to learn metagenomes embeddings

We introduce *Metagenome2Vec*, a method to transform shotgun metagenomic data into a suitable embedding representation for downstream task such as disease classification. *Metagenome2Vec* is trained from raw DNA sequences through several specific steps: metagenome embeddings are built from embeddings of reads themselves built from k-mers embeddings. We highlighted state-of-the-art algorithms that learn embeddings of k-mers and reads. Our proposed approach is implemented with two different MIL architectures:

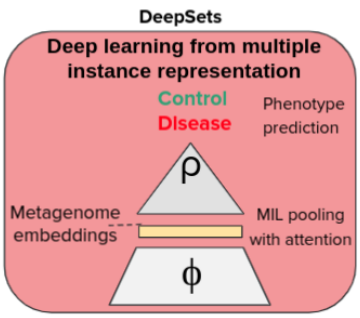
- *M2V-MIL-DS*: A MIL model based on *DeepSets* model [Zah+17] trained with a prediction loss directly for classification.
- *M2V-MIL-VAE*: A MIL model based on Variational Auto-Encoder (VAE)[1] [KW14] using a reconstruction loss to construct metagenome embeddings.

The global architecture of both models is summarized in Figure 3.2a and all blocks of the pipeline are explained below.

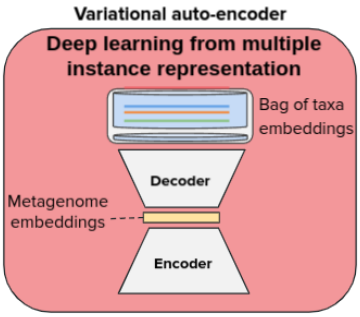
[1] We chose to implement a VAE because this model is capable of learning smooth latent state representations of the data compared to conventional auto-encoders that simply learn an encoding to reproduce the input.



(a) *Metagenome2Vec* architecture



(b) *DeepSets*



(c) *Variational Auto-Encoder*

Fig. 3.2.: Raw metagenomic data is the input of *Metagenome2Vec*. (a) All DNA sequences are embedded by *Kmer2Vec* and *Read2Vec* algorithms (Figure 3.4 and 3.8) resulting in a bag of read embeddings. (b) Then, *Read2Genome* (Figure 3.10) uses these embeddings to assign a cluster, corresponding to a genome id, for all reads. (c) Embeddings of reads in the same cluster are aggregated by summing their values. It results in a multiple instance dataset where a bag of embeddings represents one metagenome. (d) At the end, a neural network model (figures 3.2b and 3.2c) fed with multiple instance data is trained to compute metagenome representation.

3.3.1 kmer2vec: learning k-mers embeddings

3.3.1.1. Description

DNA sequences are split into several k-mers before learning k-mers embeddings. The context of a k-mer corresponds to both the preceding and the following k-mers. The context can consist of several k-mers; this parameter is called the window size and tuned to the number of surrounding context k-mers desired. After learning, all

k-mers are indexed in the embeddings matrix. Figure 3.3 illustrates this process and Figure 3.4 shows metrics to evaluate the learning.

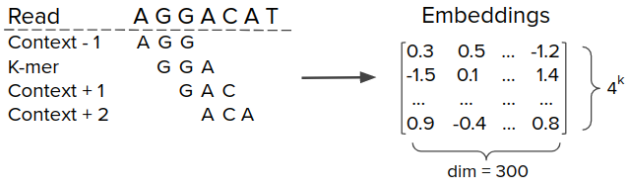


Fig. 3.3.: Left side represents a read cut into k-mers of length 3 with a window size of 2 and a padding size of 1. Right side corresponds to an embedding matrix of dimension 300 learnt with k-mers vocabulary of size $4^k = 4^3 = 64$.

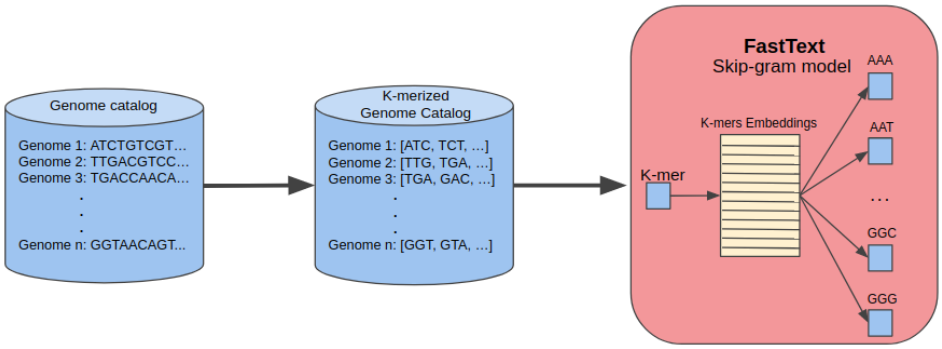
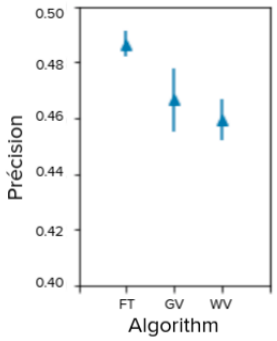


Fig. 3.4.: The preprocessing step is to transform genomes sequences into k-mers with a specific k size. On the figure k=3 because k-mers are composed of three nucleotides. Then, k-mers are passed to the *FastText skip-gram* model learning to retrieve surroundings k-mers context.

An extrinsic evaluation on a benchmark of the different kmer2vec algorithms leads to statistically significant (p-value < 0.05) better performance of *FastText* (figure and Table 3.1). That is why we mainly used *FastText* for k-mers embeddings so we described the concept and usage of this algorithm on DNA sequences.

kmer2vec	Average precision train / test
FastText	**0.60 / 0.49**
word2vec	0.56 / 0.43
GloVe	0.57 / 0.46

Average precision scores



Dispersion of the scores.
FT=*FastText*, GV=*GloVe*, WV=*word2vec*.

Tab. 3.1.: Extrinsic evaluation of kmer2vec algorithm with $k = 6$ on a *Read2Genome* task (section 3.3.3): classification of reads into 10 species (122k reads on train and validation) from a simulated dataset with balanced species abundances. K-mer embeddings are averaged and fed to a multiple layer perceptron classifier trained with 20 k-fold cross validation.

We described the three algorithms used in our experiments *word2vec*, *FastText* and *GloVe*:

word2vec This is the well-known model popularized by Google by Mikolov et al. [Mik+13]. It is a shallow two-layer neural network auto-encoder. We opted for the *skip-gram* version of the model. So, the neural network based on a similar architecture has been trained to predict the most obvious surrounding context for each k-mers. The prediction is based on the softmax function that gives the posterior distribution of k-mers:

$$p(c|k) = \frac{exp(s(k,c))}{\sum_{c_i=1}^{|K|} exp(s(k,c_i))}$$

$$s(k,c) = v_c^T v_k$$

With k the current k-mer, K the vocabulary of k-mers, c the context k-mer, v represents a vector, $s(w,c)$ denotes the scoring function between a word vector w and a context vector c.

FastText Published by Facebook [Jou+16; Boj+16], this algorithm brings the concept of *subword* information based on character n-grams. In fact, *FastText* works like the *skip-gram word2vec* model but with a different scoring function. In *word2vec*

a dot product is done between the current word vector and its context word vectors. Here, the dot product is computed between all corresponding characters from min-n-gram to max-n-gram (two hyper parameters). For example, if we consider a k-mer "ATACCA", min-n-gram=3 and max-n-gram=6, the n-grams are {ATA, TAC, ACC, CCA, ATAC, TACC, ACCA, ATACC, TACCA, ATACCA}. A k-mer is finally represented as the sum of the vector representations of its n-grams. The scoring function is re-written as follows:

$$s(k,c) = \sum_{g \in \mathcal{G}_k} z_g^T v_c$$

Where $\mathcal{G}_k$ is the set of character n-grams of the k-mer k, z is a vector representation of all character n-grams and v is a k-mer vector. This method allows a reliable representation to be learnt even for rare k-mers. It can be useful since we don't know the true size of each k-mer and where they are separated.

GloVe : Proposed by Standford researchers [PSM14], it computes a matrix factorization of the co-occurrence matrix. It aims to minimize the following cost function:

$$\sum_{i=1}^{V} \sum_{j=1}^{V} f(X_{ij})(v_{k_i}^T v_{k_j} + b_{k_i} + b_{k_j} - log(X_{ij}))^2$$

$$f(x) = \begin{cases} (x/x_{max})^\alpha & if(x < x_{max}) \\ 1 & otherwise \end{cases}$$

Where X_{ij} is the co-occurrence between the current k-mer k_i and the context k-mer k_j. v and b are respectively the vector and the bias of the k-mers. f adds a weight determined by the corresponding co-occurrence value and a distribution in terms of α. It prevents giving high values to common k-mer pairs with the threshold x_{max}.

We have seen that these three word embeddings algorithms could be easily adapted to DNA sequences by replacing words by k-mers. Any state-of-the-art word embedding algorithms could be potentially included in our architecture to be tested beside the three that we tested.

3.3.1.2. Analysis: k-mer embeddings intrinsic evaluation

K-mer embeddings are trained in a self-supervised manner where the algorithm tries to predict the surrounding k-mers regarding the current one. The three main adjustable hyper parameters in these algorithms are the size of the embeddings (dimensionality complexity), the k-mer size (smaller or bigger pieces of DNA) and the window size (more or less surrounding words). It creates a large parameters space

that influences severity points like the vocabulary size, the embeddings learning, the processing time and more globally the final representation. Increasing the value of k leads to an increase of the volume of the dictionary and the learning time inevitably becomes longer for any algorithm (Figure 3.5 and Table 3.2).

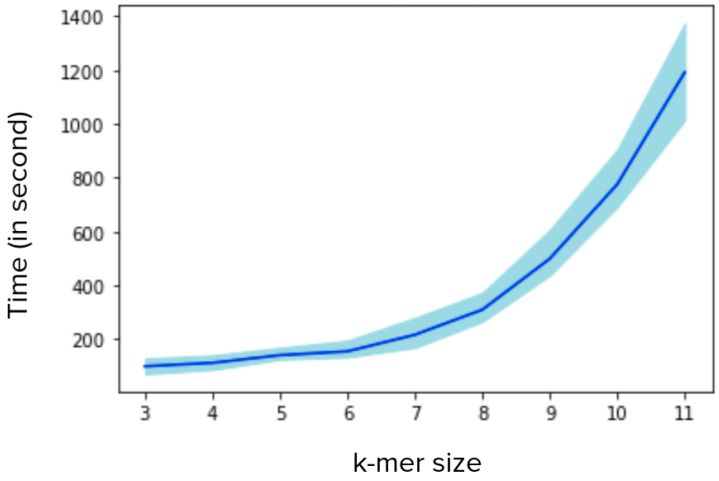


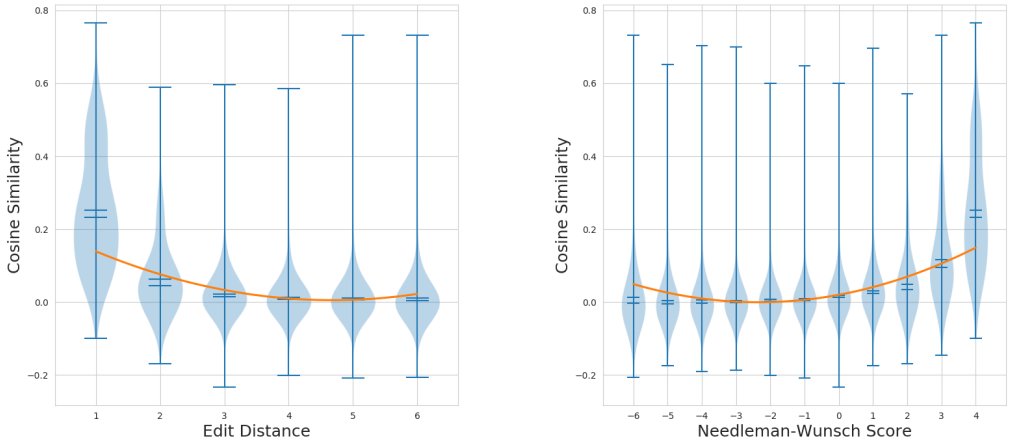
Fig. 3.5.: Execution time depending on k. *FastText* trained 5 times on 15 cpus for each k on a small dataset (30k genes)

	FastText	***word2vec***	***GloVe***
$k = 3$	$\simeq 20m$	$\simeq 1h$	$\simeq 10m$
$k = 6$	$\simeq 3h$	$\simeq 5h$	$\simeq 1h$
$k = 9$	$\simeq 2d$	$\simeq 2d$	$\simeq 1.5d$

Tab. 3.2.: Execution time of *FastText*, *word2vec* and *GloVe* depending on k size, trained on a dataset of 5M genes. The other hyper parameters for each algorithm are set by considering those recommended. Models are multi-threaded over 50 cpus.

Analyzing k-mers vectors and finding best hyper parameters are done by intrinsic evaluation of the embeddings. The intrinsic evaluation is an important test that could help to identify whether the algorithms learned good DNA embeddings. Unfortunately, this task is not obvious depending on the DNA sequences. There is not a lot of information about the vocabulary compared to natural language where we assume that the vectors of two words like synonyms have a high similarity. Several intrinsic evaluation methods for NLP word embeddings are enumerated in Bakarov [Bak18] (such as "word semantic similarity", "word analogy" or "synonyme detection") but none of them can be used with DNA because they rely on text-specific concepts. To overcome the fact that these methods are not available to evaluate the DNA embeddings, distance between k-mer chains is taken into account. Ng

[Ng17] measures the relation between the cosine similarity of two vectors with their corresponding k-mers Needleman-Wunch score [NW70]. In Min et al. [Min+17], authors prefer to compute the relation between the cosine similarity and the Edit distance. Both Edit distance and Needleman-Wunsch scores are computed on k-mer chains and compared to the cosine similarity of their embeddings. Figure 3.6a and 3.6b confirm that the distance between k-mers and between their embeddings do correlate. Unfortunately, these methods are only feasible when k is not too high, generally less than or equal to 6. Indeed, when k increases, so does the number of k-mers in vocabulary, which makes the calculation of distances much too long.



(a) Correlation between Edit distance of k-mers and cosine similarity of k-mers embeddings

(b) Correlation between Needleman-Wunsch score of k-mers and cosine similarity of k-mers embeddings

Fig. 3.6.: Each violin plot shows mean, median and the extreme values at a specific score. A smaller Edit distance and a higher Needleman-Wunsch score implies that k-mers are more similar. A higher cosine similarity implies that vectors are more co-linear.

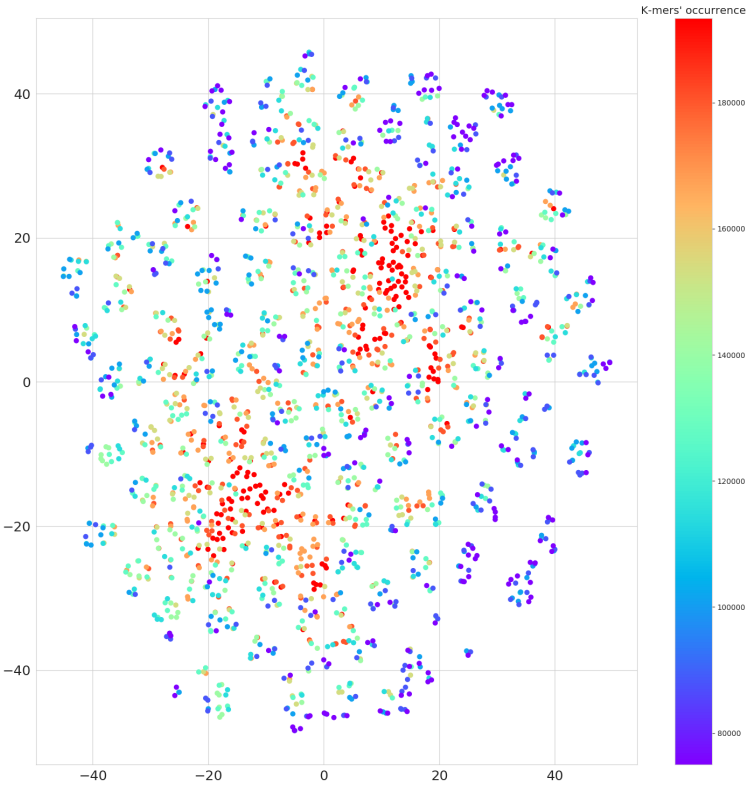


Fig. 3.7.: Each point is the projection into a 2D space with the *t-SNE* algorithm of genome embeddings from the *FastDNA* model. Points similarly colored have the same family.

We have seen different algorithms allowing to create k-mer embeddings by projecting them into a latent space. Since the reads are composed of several k-mers, we are now interested in creating read embeddings.

3.3.2 read2vec: learning read embeddings

3.3.2.1. Description

It has been shown that the algorithms constructing word embeddings give good results for representing short sentences by simple arithmetic operations on word vectors [Mik+13; PSM14; Jou+16]. Nevertheless, more sophisticated approaches for sentence embeddings have been developed and obtan even better results. Sequence-to-sequence models [SVL14] for instance, use a first network called the *encoder* to encode the sentence information. And a second one, called *decoder* decodes the sentence information for a specific task such as a translation where the authors

have demonstrated great performance. The hidden layers of the encoder represent the sentence embeddings. *Skip-thought* [Kir+15] vectorizes sentences with this approach, learning to generate surrounding sentences. To gain computation time compared to *skip-thought*, Hill et al. [HCK16] have proposed *fastSent*, an additive log-linear sentence model using a bag of words embedding to represent sentences. *SDAE* [HCK16], uses a LSTM encoder-decoder to reconstruct noisy sentences (words are removed or switched places according to a probability). More recently, *BERT* model [Dev+18] trains an encoder with a self-attention mechanism [LPM15], called transformer, to learn contextual relations between words to retrieve masked words or predict the next sentence. DNA sequences can be transformed into embeddings using previous algorithms in the same way as sentences are.

Conneau et al. [Con+17] proposed a supervised neural network named *InferSent* based on two parts. The network begins with an encoder creating sentence vectors. Then the deep representation passes into a *natural language inference* [2] classifier predicting one of the three NLI labels. Some other approaches developed are training-free models. For example, *SIF* algorithm [ALM17] embeds sentences by summing its pre-trained word embeddings weighted with the reverse document frequency. Then they subtract from the sentence embedding the first principal component of the matrix. *p-mean* [Rüc+18] also demonstrated good results by averaging and concatenating power means of the embeddings. The Table 3.3 resumes specificities of these algorithms.

Sentence embedding	Supervised	Ordered sentences	training-free
Skip-Thought [Kir+15]	-	✓	-
FastSent [HCK16]	-	✓	-
BERT [Dev+18]	-	-	-
SDAE [HCK16]	-	-	-
InferSent [Con+17]	✓	-	-
SIF [ALM17]	-	-	✓

Tab. 3.3.: Sentence embedding algorithms and their specificities

In our experiments, three algorithms were integrated in the workflow:

- *FastText* (see section 3.3.1 for more details) sentence embeddings as implemented in the package

[2] Natural language inference is the task of determining whether a "hypothesis" is true (entailment), false (contradiction), or undetermined (neutral) given a "premise"

- *FastDNA* sequence embeddings extracted after the model were trained to classify taxonomy reads at the species level
- *Transformer*, A sequence-to-sequence base transformer applied on language modeling task [3].

Each of these algorithms respects two properties: (i) being efficient enough to process the millions of sequences per metagenome (a non-blocking point in theory but important for implementation), and (ii) not involving sentence order in the prediction task. Figure 3.8 gives an example of *Read2Vec* with a transformer.

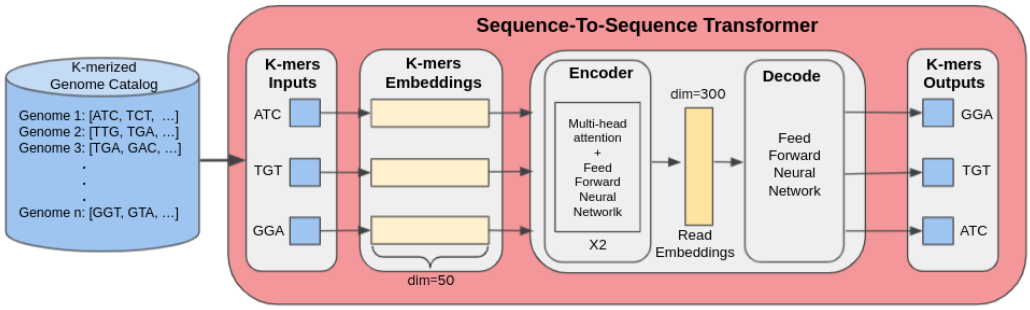


Fig. 3.8.: *Read2Vec* architecture with a *Transformer*. Sequences, cut into k-mers, pass into a transformer sequence-to-sequence language model. A first layer converts k-mers to their embeddings learnt in *Kmer2Vec* (Figure 3.4). The encoder creates the read embeddings with two blocks composed by a multi-head attention and a feed forward neural network. The decoder tries to predict the next k-mers from the source sequence passing the read embeddings in a fully connected layer before computing the softmax to get a probability for each k-mers. When k is relatively big, this last layer is quite intensive to compute because its complexity grows linearly with the size of the vocabulary. Thus, the adaptive softmax proposed by Grave et al. [Gra+17] is used instead of softmax to be more efficient without reducing performance.

3.3.2.2. Analysis: Read embeddings intrinsic evaluation

Embeddings at the read level cannot benefit from analysis (correlation between Edit distance or Needleman-Wunsch score and k-mers embeddings) in section 3.3.1.2 because reads' length are a lot bigger than k-mers. Nevertheless, as a genome catalog has been used to train the read embeddings, genomes can be projected in this new vector space. We would expect that species from the same genus or with a similar genetic material are more closely related to each other in the embedding space. We have thus set up two methods to quantify this phenomenon. One is to project

[3] Language modeling algorithms determine the probability distribution for the likelihood of a given word (or a sequence of words), to follow a sequence of words

and visualize genome embeddings using the t-SNE algorithm. Results on Figure 3.9 highlight that some clusters are formed of genomes from the same family. The other method is to compute a Mantel test and compare the correlation between two distance matrices of genomes. The first is the cosine similarity between genome embeddings, the second is the Mash distance which is a genome distance estimation using the MinHash algorithm between genome DNA[4]. A high value in the mantel test implies that cosine similarity of the embeddings is correlated with the mash distance of DNA, then it gives a good indicator on the relevance of the representation learnt by the model. Models are tuned and results are reported in Table 3.4.

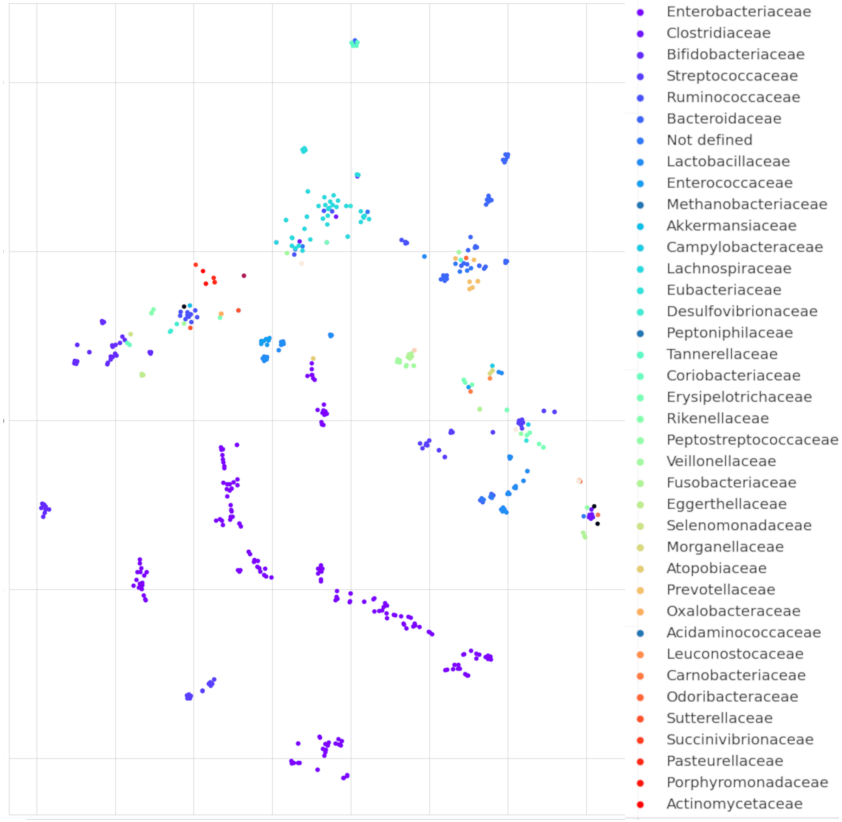


Fig. 3.9.: Each point is the projection into a 2D space with t-SNE algorithm of genome embeddings from FastDNA model. Points similarly colored have the same family.

[4]It is computed using the GitLab from the study of Criscuolo [Cri19] implementation.

Method	Mantel Test
FastText k=3,dim=300	0.50
FastText k=6,dim=300	0.54
FastText k=9,dim=300	0.56
FastText k=13,dim=100	0.61
Transformer k=3,dim=300	0.52
Transformer k=6,dim=300	0.56
Transformer k=9,dim=300	0.59
FastDNA k=13,dim=100	**0.64**

Tab. 3.4.: Mantel Test scores between Mash distance and genome embeddings. Parameters k and dim refer to k-mer size and embeddings dimension respectively. Best value is obtained for *FastDNA*

FastDNA has the highest scores in this analysis. *Transformer* has better results than *FastText* for similar k. However, transformer could not benefit from learning with a bigger k due to the complexity and the memory footprint of the model increasing exponentially with the size of the vocabulary. This is a blocking factor because it is shown that increasing k leads to higher scores on the Mantel test. After analyzing the vector embeddings with intrinsic evaluation, it is not yet possible to assert if the embedding representation obtained by the algorithms performs well on specific tasks. Extrinsic evaluation is used over all the representations to determine which approach is best suited. An extrinsic evaluation of the embedding is based on the results of its use for a specific purpose. In both steps *Read2Genome* and *Metagenome2Vec*, DNA embeddings are handled to perform read classification and disease classification respectively. Thus, extrinsic evaluation is also computed on the different learnt embeddings by analyzing prediction results reported in section 3.3.3.2 for *Read2Genome* and 3.4.2 for *Metagenome2Vec*.

3.3.3 read2genome: reads classification

3.3.3.1. Description

A metagenome is composed of millions of reads which represent portion of DNA genomes. Quantitative metagenomics focus on retrieving the origin of each read with bioinformatics tools. Several studies combining metagenomics and deep learning also aim at projecting raw sequences to classify them into a taxonomic level (section 3.2.2). We would like to take advantage of the putative origin of the reads

to construct metagenome representation. *Read2Genome* then acts as a clustering process producing bag of reads with genome similarity. To address the question of predicting the genome to whom a read most probably belongs, we have relied on two methods in our experiments. Firstly, *FastDNA* [MV19] that learns embeddings and classification with an end-to-end supervised model. Secondly, a Multi-Layer Perceptron (MLP) classifier that takes as inputs the read embeddings learnt by self-supervised training with *Read2Vec*. Figure 3.10 summarized the *Read2Genome* architecture.

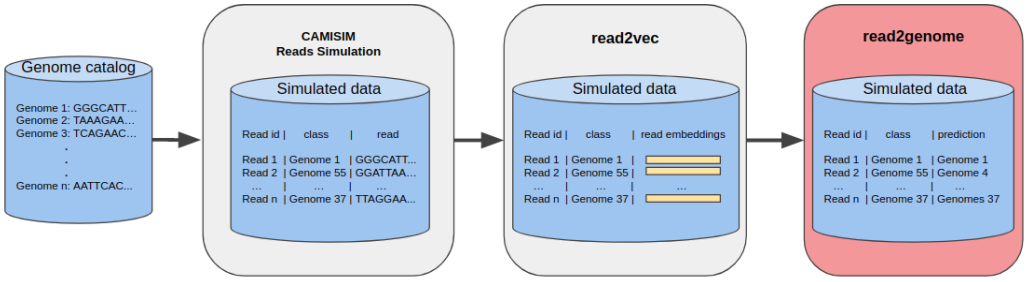


Fig. 3.10.: A catalog of complete genomes is used by the *CAMISIM* software [Fri+19] to simulate metagenomic data with a specific taxonomic profile (abundance of species). The resulting dataset is a set of reads associated with the identifier of the genome from which they originate. Reads are embedded by *Read2Vec* (Figure 3.8) before being passed into *Read2Genome* trained to retrieve their source genome.

To calculate a metagenome embedding, we started with a basic method averaging all the embeddings of k-mers present in a metagenome (*vectorial representation* in section 3.3.4.1) like embeddings of sentences are built by averaging embedding of words. But this representation is too "brutal" and too much information is lost to give conclusive results, in our experiments, in terms of phenotypic stratification of patients. Therefore, we searched for in a method that allows both, using the predicted classes (taxa) of reads, to group the embeddings of reads of the same predicted class sharing common concepts and to estimate an abundance table of a metagenome.

3.3.3.2. Analysis: Read embeddings extrinsic evaluation on a read classification task

As shown in Figure 3.10 and in section 2.2.1, the datasets used in the experiments are composed of simulated metagenomes. The simulation allows to represent the NGS sequencing data while keeping the label of the reads allowing the training

of supervised models. The evaluation of *Read2Genome* is done with both types of sequencing technology Illumina and Nanopore.

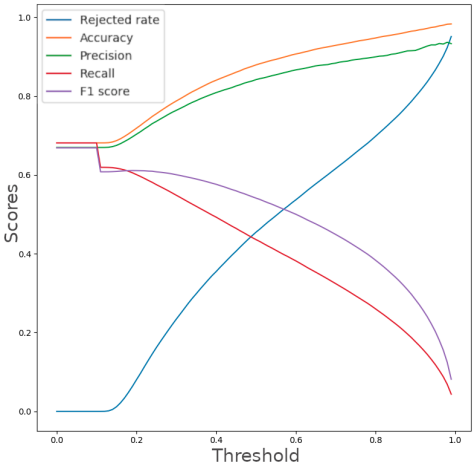
Given a read, we ensure that the *Read2Genome* model returns probabilities associated with the prediction that it belongs to a genome. In This way, we can set a threshold to reject uncertain classifications. As there is a extremely high number of sequences by metagenome (often greater than 10M), rejecting uncertain predictions improves the precision of the model without impacting clusters of reads. Liang et al. [Lia+20] also uses a reject threshold determined manually in *DeepMicrobes*. The metric for controlling the reject is the "rejection rate" calculated by dividing the number of rejected reads by the total number of reads.

We compare the results of *FastDNA* and *Transformer+MLP* models on Illumina datasets trained over 10 of the 235 species in the dataset (Only 10 species have been selected to allow rapid learning). As *FastDNA* obtains the best scores on 10 species, we trained the model on the whole 235 species with parameters recommended by Menegaux and Vert [MV19] which are set to 13 for k-mer size, 100 for embedding dimension and 30 for the number of epochs. We computed and plotted the accuracy, precision, recall, f1-score and rejected rate in accordance with the rejected threshold (see Figure 3.11). The threshold axis corresponds to the minimum probability of the class predicted by the model so that the read is not rejected. Metrics' formulas are recalled below:

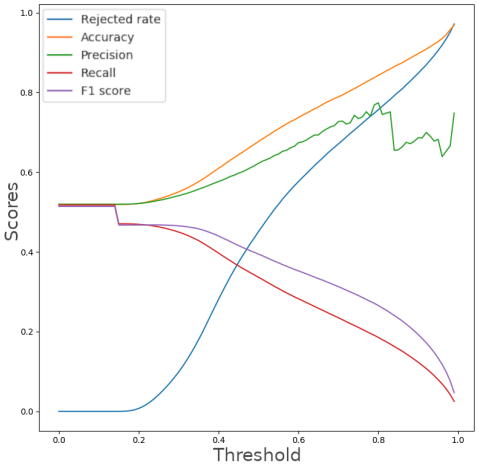
$$Accuracy = \frac{TP+TN}{N}, \quad Precision = \frac{1}{C}\sum_{i}^{C}\frac{TP_i}{TP_i+FP_i}, \quad Recall = \frac{1}{C}\sum_{i}^{C}\frac{TP_i}{TP_i+FN_i}$$

$$F1\text{-}Score = 2 \times \frac{precision \times recall}{precision+recall}, \quad Rejected\,Rate = \frac{R}{N}$$

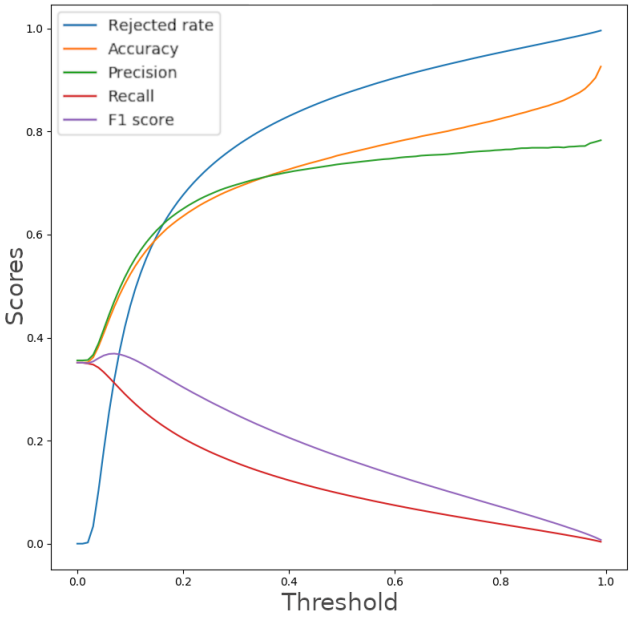
With $N = \#$ of samples, $C = \#$ of classes, $R = \#$ of rejected samples, $TP =$ True Positive, $TN =$ True Negative, $FP =$ False Positive, $FN =$ False Negative



(a) *FastDNA* on 10 species



(b) *Seq2Seq+MLP* on 10 species



(c) *FastDNA* on 235 species

Fig. 3.11.: Scores obtained by Illumina reads classification into species from the 1.59M simulated reads of the validation dataset. The higher the threshold, the better the accuracy and the lower the recall.

These curves give information on the impact of the prediction rejection threshold allowing to adapt it according to the expected rejection percentage and classification score. We tested *fastDNA* algorithm on the simulated Nanopore reads to compare the result with longer read in the task of binning (Figure 3.12)

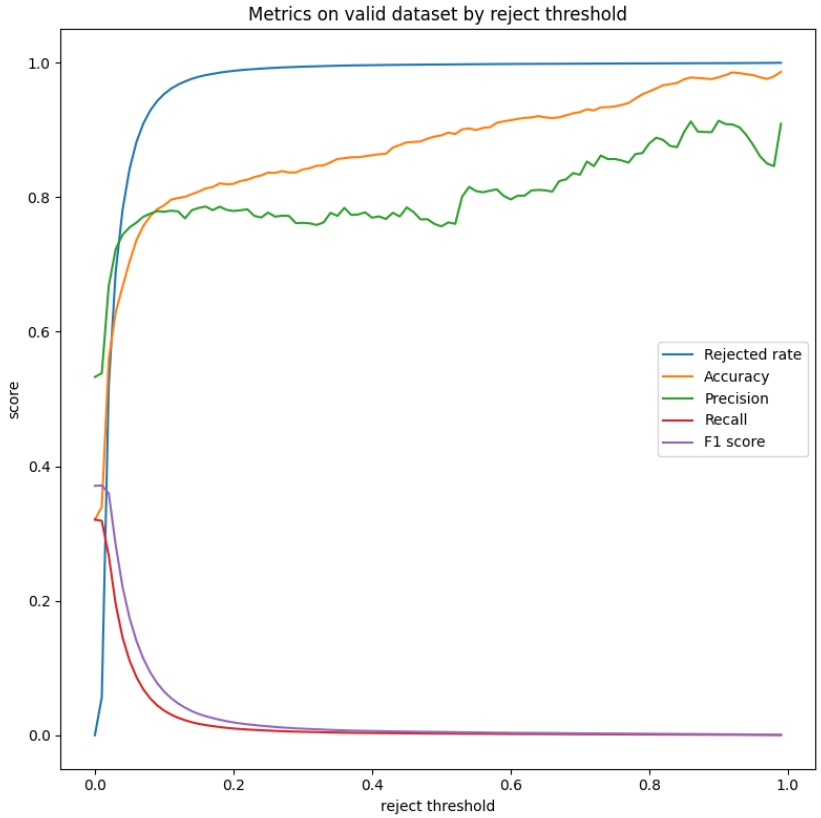


Fig. 3.12.: Scores obtained by Nanopore reads classification into species with *FastDNA* from the 315K simulated reads of the validation dataset. The higher the threshold, the better the accuracy and the lower the recall.

We can note that, on Nanopore simulated dataset, increasing the threshold will induce a non-negligible growth of the rejection rate which can be problematic if too many reads are not retained. We manually select good trade-off between accuracy and rejection rate for the following experiments. *Read2Genome* models trained from Illumina reads have a rejection threshold set to 0.2, while those trained from Nanopore data have no defined rejection threshold (it is equal to zero).

3.3.4 metagenome2vec: learning metagenome embeddings

The following step is to create metagenome embeddings using read embeddings or a set of reads embeddings. We propose to consider two different approaches in building metagenome embeddings: (i) the vectorial representation as baseline and (ii) the MIL representation as our reference method. The notations in the next sections are in accordance with those introduced in section 3.1.

3.3.4.1. metagenome2vec: Vectorial representation

Once a low-dimensional representation of the reads is available, all reads from a given metagenome are transformed into embeddings. In this representation, called *M2V-VR*, they are all summed together, resulting into a single instance embedding for one metagenome. A representation of metagenome can be computed as shown in the equation 3.1:

$$\Phi(x_m) = \sum_{s \in x_m} \omega(s)\,, m \in M,\ \omega : \begin{cases} x_m \to E \\ s \mapsto \omega(s) \end{cases} \tag{3.1}$$

With ω the *Read2Vec* transformation, x_m the ensemble of reads in the metagenome m and E the dimension of the embeddings.

The vectorial or tabular representation is the one used by most ML algorithms. It relies on a more abstract representation than the multiple instances representation. Indeed, all the information of an entire metagenome, its millions of reads related to hundreds of different genomes, is summarized into a unique vector in the latent embeddings space.

In addition to the classification evaluation, we calculated and evaluated a clustering based on the embedding representation. For this clustering, m_1 metagenomes are selected and $m_2 < m_1$ others are cut in 10 sub parts. Each metagenome, or part of the metagenome, is represented by one vector. An agglomerative clustering is trained on these embeddings to compute a cluster map and show visualize between clusters (Figure 3.13). Results show logically that embeddings from portions of the same metagenome are closer to each other. They also indicate that metagenomes of the same class are more likely to be found in the same cluster.

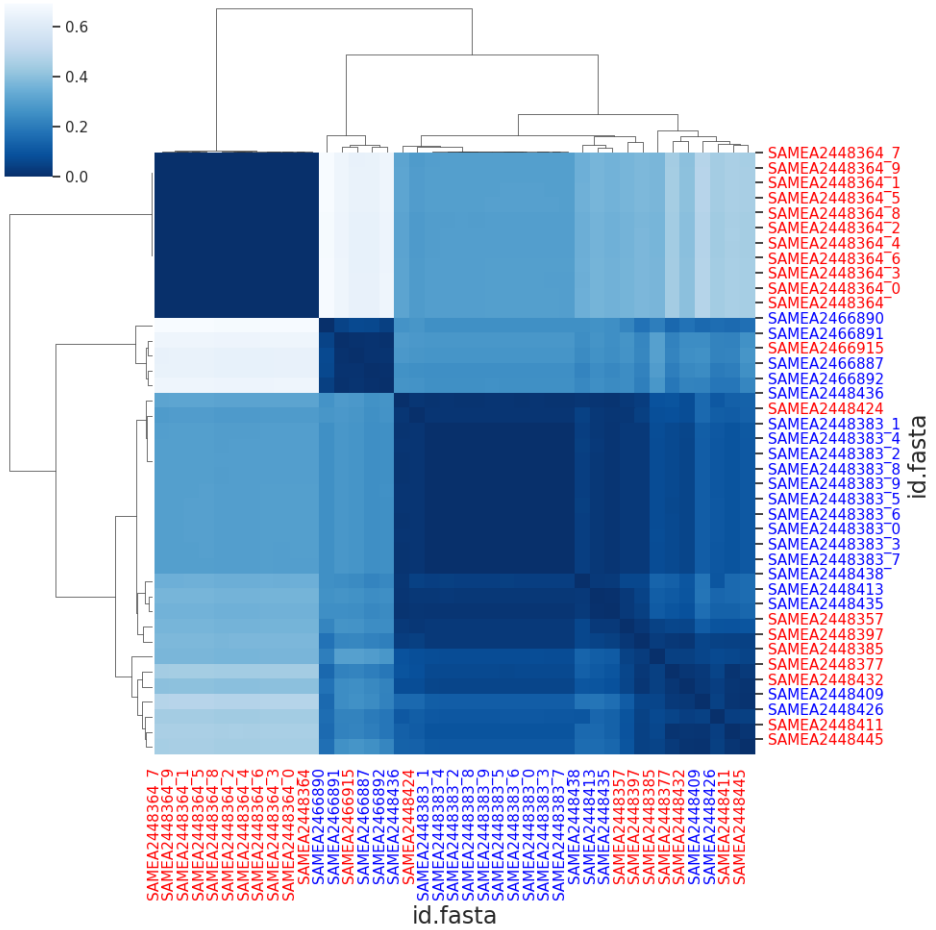


Fig. 3.13.: Cluster map computed on the Colorectal Dataset with the *Metagenome2Vec* vectorial representation. Blue ids and red ids refer to healthy patients and sick patients respectively. Underscores on ids followed by a digit correspond to partitions of the same metagenome. On the map, the darker the color, the more similar the metagenomes.

3.3.4.2. metagenome2vec: Multiple instance learning representation

Metagenomic data can be thought of as a set that contains millions of reads representing one sample (Figure 1.7). The size of a set can vary depending on the sequencing technology and there is no specific order between reads within a set. Learning from such bags of reads correspond to a Multiple-Instance Learning problem that deals with an objective function that is invariant to permutation and operates on non-finite dimensional vectors [Zah+17]. In this thesis, we implement deep learning algorithms from the work of Zaheer et al. [Zah+17] named *DeepSets* with an extension of the multiple instance layer including an attention mechanism from Ilse et al. [ITW18] and a Variational Auto-Encoder [KW14] (see Figure 3.14

and appendix A.1 for more details about *DeepSets* and Figure 3.15 for the VAE). The attention mechanism assigns a weight for each instance to determine which one in the set helped to predict the label. As the metagenomes are represented as a bag of genome embeddings, it is interesting to integrate such an aggregation operation to determine the taxa that play a bigger role in the prediction. On the other hand, the VAE is more adapted to the construction of vector representation because it learns to reconstruct the training data and not to classify them directly. As a result, the model's weights can be reused for other datasets without having to be learnt again.

Instead of aggregating all the computed read embeddings to form one vector, the first idea is to keep this representation to save all information. Unfortunately, one metagenome is composed of potentially millions of reads. Thus, a bag with all these reads is far too large to fit in memory for further processing by the ML algorithms. We advocate another approach, consisting of first training a classifier (*Read2Genome*) to predict the genomes from which the DNA sequences may have originated. Rather than summing all read embeddings as in the previous method, it is possible to sum embeddings of reads belonging to the same taxonomic levels, namely species or genus. As a result, each metagenome is represented by a set of taxa embeddings.

$$\Psi(x_m) = [\Phi(c_1), \Phi(c_2), ..., \Phi(c_n)] \, , c_{1..n} \in \mathrm{C}(x_m) \tag{3.2}$$

Where C is the *Read2Genome* function clustering reads of a metagenome into n clusters, thus c_n is a group of reads, x_m the ensemble of reads in the metagenome m and $c_n \subset x_m$.
As reads are fragments of DNA from several genomes, grouping them into clusters projected onto the embedding vector space could bring specific information for each metagenome.

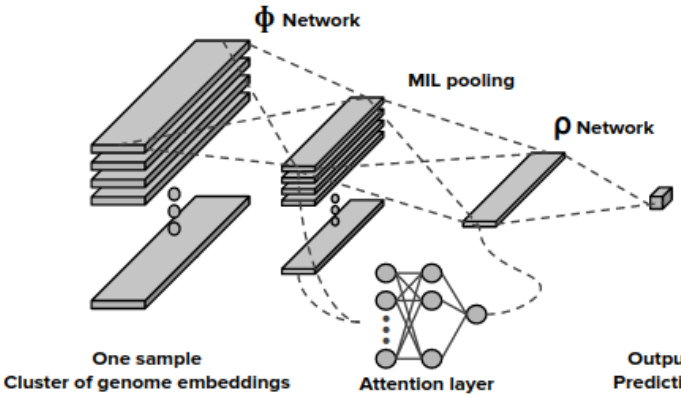


Fig. 3.14.: *M2V-MIL-DS*: DeepSets neural network architecture with attention as MIL layer [Zah+17; ITW18]. The input is a set of genome cluster embeddings and the output is the phenotype prediction.

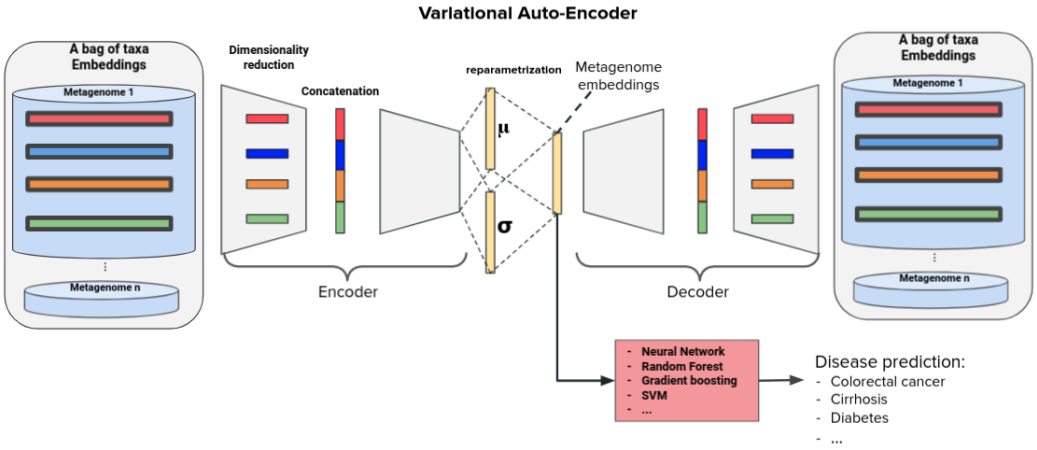


Fig. 3.15.: *M2V-MIL-VAE*: A variational auto-encoder where the encoder takes as input the bag of taxa embeddings that is passed to fully connected (FC) layers to reduce the dimensionality of the embeddings. Then, all embeddings in the bag are concatenated before being passed again to FC layers encoding a distribution over the latent space with μ (mean) and σ (variance) vectors. Next, a reparametrization step allows to back propagate the sampling gradient error by defining the final embeddings with the following formula: $z = h(x) + g(x),\ Z \sim \mathcal{N}(0, \mathcal{I})$. Where $h(x)$ computes σ, $g(x)$ computes μ and $\mathcal{N}$ represents the normal distribution. Finally, the decoder takes the embeddings and applies transposed operations to decompose the condensed representation trying to find the original bag of taxon embeddings. The condensed representation (in yellow) is fed to a classification model for learning disease prediction. This last step can be accomplished by fine-tuning in order to relearn the model weights based on a specific classification task.

3.4 Experiments and Results

We devised several experiments to test the efficacy and the efficiency of our novel *Metagenome2Vec* algorithm to classify metagenomes onto classes of diseases with which the hosts are associated. The performance of *Metagenome2Vec* w.r.t. the state-of-the-art was tested on four benchmark disease classification tasks (section 2.1) as well as a simulated dataset (section 2.2.2)). Moreover, to understand the source of power of the *Metagenome2Vec* algorithm, we also tested the intrinsic quality of the learnt embeddings and the ability to assign a read to the right taxa. The next sections are organized as follows: 3.4.1 provides information on reference methods and 3.4.2 highlights evaluation methods and metrics for disease prediction.

3.4.1 Reference Methods compared to metagenome2Vec

To our knowledge there is no other study applying machine learning directly on raw metagenomic data to predict disease. In general, disease classification with metagenomic data is done with standard pipeline using species-level relative abundances and presence of strain-specific markers [Pas+16]. On top of these bioinformatic processes, ML algorithms like SVM, Random Forest or Elastic Net are trained to make predictions. More recently, Oh and Zhang [OZ20] have proposed to highlight the use of auto-encoder models on such metagenomic abundance tables. Results are reported in Table 3.5 and are used in this paper as part of the state-of-the-art benchmark.

3.4.2 Results of the Disease prediction tasks

The computing resources used for the experiments are between 1 up to 6 nodes with 24CPUs each and 1 up to 2 GPUs NVIDIA Tesla P100 or NVIDIA Quadro K5200. All experiments are done by limiting the number of reads to ten million; the training time can take 2 to 5 days and there are many parameters to test for both structuring and machine learning. The inference execution time is about 1 hour for a metagenomic sample. The datasets are composed of only hundreds of samples so to tune the hyper parameters and avoid overfitting we apply a nested cross validation. In that way, 20% of the data form the test set, the 80% remaining are used to *Metagenome2Vec* form a 10-fold cross validation to tune hyper parameters with 20% of the data as validation set. The whole operation is repeated 10 times with different

train and test sets. The tuning is driven by accuracy score. AUC, precision, recall and F1 score are also computed.

DL models, for the *Metagenome2Vec* MIL representation, are trained with the following techniques to prevent the neural networks to overfit:

- Drop out: remove randomly picked weights in order to force the model to re-train some of its parameters.
- Scheduler decay learning: decrease the learning rate to make a better convergence.
- L2 penalty: add penalty to the loss function to ensure better generalization.

Methods *MetaML* [Pas+16] and *DeepMicro* [OZ20] are the reference methods. Bag of K-mers (*BoK*) method is related to Bag of Word (*BoW*) and has been computed with three different values of k equal to 3, 6 and 9[5]. Thus, *BoK* represents a metagenome as a vector by counting all the occurrences of its k-mers without using embeddings, this is the baseline to confirm embeddings usefulness. *M2V-VR* is *Metagenome2Vec* vectorial representation, *M2V-Abundance* represents the species abundance table computed by the *Read2Genome* stage and M2V-MIL is "*Metagenome2Vec* multiple instance learning representation". *Read2Vec* and *Read2Genome* are trained by the *FastDNA* [MV19] algorithm with a k-mer size equal to 14 and an embeddings dimension equal to 50 (recommended by authors). *M2V-MIL-DS* and *M2V-MIL-VAE* models use *DeepSets* and *VAE* deep learning architecture respectively; they are trained and evaluated on the MIL representation. Models used for BoK, *M2V-VR* and *M2V-Abundance* methods are tuned with random search using 100 different sets of parameters. *M2V-MIL-DS* and *M2V-MIL-VAE* are tuned with approaches more adapted for model composed by a lot of hyper parameters like neural network. Thus, Bayesian optimization is applied for continuous parameters and bandit optimization for discrete parameters.

Datasets

- Illumina: 4 real world datasets named *Colorectal*, *Cirrhosis*, *Obesity* and *T2D* that refer to Table 2.2 are benchmarked with all previous methods. The *Null Model Illumina* from Table 2.3 is used to evaluate *DeepSets* efficiency and interpretability.

[5] It cannot be computed with a higher value of k due to the number of distinct k-mers in the vocabulary that becomes too large.

- Nanopore: Both *Null Model Nanopore* and *Ecological Nanopore* from Table 2.3 are benchmarked to evaluate the efficiency of *Metagenome2Vec* on Nanopore data.

Results The tables 3.5 and 3.6 summarize our results. Accuracy, F1-score, Precision, Recall and AUC are the computed metrics. The standard deviation is calculated for the accuracy score and written with the symbol ±. All the scores are the best of all the experiments

Method	Metrics	*Colorectal*	*Cirrhosis*	*Obesity*	*T2D*	*Null Model Illumina*
MetaML	Accuracy	0.81 ±0.068	0.88 ±0.043	0.64 ±0.028	0.66 ±0.052	-
	precision	0.82	0.89	0.54	0.67	-
	recall	**0.81**	0.88	0.64	0.66	-
	F1-score	**0.79**	0.88	0.54	0.66	-
	AUC	**0.87**	**0.95**	0.66	0.74	-
DeepMicro	AUC	0.81	0.94	0.66	0.76	-
BoK k=3 / k=6 / k=9	Accuracy	0.64 / 0.69 / 0.65 ±0.061 / ±0.055 / ±0.071	0.67 / 0.73 / 0.75 ±0.077 / ±0.062 / ±0.043	0.64 / 0.65 / 0.65 ±0.028 / ±0.032 / ±0.035	0.58 / 0.61 / 0.62 ±0.047 / ±0.076 / ±0.071	0.61 / 0.66 / 0.63 ±0.032 / ±0.040 / ±0.057
	precision	0.81 / **0.88** / 0.84	0.66 / 0.76 / 0.77	0.64 / 0.65 / 0.65	0.57 / 0.59 / 0.61	0.61 / 0.66 / 0.71
	recall	0.57 / 0.61 / 0.60	0.77 / 0.69 / 0.74	0.96 / 0.95 / 0.95	0.53 / 0.58 / 0.58	0.56 / 0.62 / 0.55
	F1-score	0.55 / 0.59 / 0.56	0.70 / 0.72 / 0.74	0.77 / 0.78 / 0.78	0.55 / 0.58 / 0.61	0.59 / 0.64 / 0.65
	AUC	0.66 / 0.73 / 0.67	0.71 / 0.77 / 0.77	0.53 / 0.53 / 0.54	0.61 / 0.63 / 0.65	0.65 / 0.69 / 0.67
M2V-VR	Accuracy	0.72 (0.058)	0.79 (0.044)	0.65 ±0.008	0.66 ±0.050	0.85 ±0.047
	precision	0.77	0.78	0.65	0.70	0.87
	recall	0.64	0.67	0.89	0.56	0.81
	F1-score	0.69	0.70	0.79	0.60	0.84
	AUC	0.79	0.81	0.66	0.70	0.85
M2V-MIL-DS	Accuracy	0.81 ±0.065	0.82 ±0.056	0.66 ±0.022	0.68 ±0.059	0.92 ±0.070
	precision	0.78	0.83	0.66	0.68	0.94
	recall	0.76	0.84	0.89	0.63	0.91
	F1-score	0.76	0.83	0.79	0.65	0.92
	AUC	0.81	0.83	0.62	0.71	0.92
M2V-MIL-VAE	Accuracy	0.81 ±0.063	0.85 ±0.049	**0.74** ±0.036	0.76 ±0.055	-
	precision	0.80	0.84	**0.77**	0.73	-
	recall	0.70	0.80	0.84	0.75	-
	F1-score	0.74	0.81	**0.83**	0.74	-
	AUC	0.78	0.84	**0.68**	0.78	-
M2V-Abundance	Accuracy	**0.82** ±0.096	**0.90** ±0.052	0.71 ±0.036	**0.80** ±0.046	-
	precision	0.80	**0.92**	0.74	**0.78**	-
	recall	0.74	**0.89**	**0.85**	**0.82**	-
	F1-score	0.75	**0.90**	0.79	**0.80**	-
	AUC	0.82	0.94	0.63	**0.83**	-

Tab. 3.5.: Classification metrics of four real-world datasets (*Colorectal*, *Cirrhosis*, *Obesity*, *T2D*) and one simulated dataset (*Null Model Illumina*). Results are reported for two reference methods (*MetaML* and *DeepMicro*) that use species-level relative abundances and presence of strain-specific markers. *BoK*, *M2V-VR*, *M2V-MIL-DS*, *M2V-MIL-VAE* and *M2V-Abundance* are our methods tested in this experiments.

Method	Metrics	*Null Model Nanopore*	*Ecological Nanopore*
M2V-VR	Accuracy	0.80 ±0.029	0.51 ±0.030
	precision	0.85	0.51
	recall	0.75	0.49
	F1-score	0.79	0.50
	AUC	0.87	0.51
M2V-MIL-VAE	Accuracy	**0.93** ±0.027	0.67 ±0.22
	precision	**0.94**	0.67
	recall	**0.93**	0.64
	F1-score	**0.94**	0.66
	AUC	**0.97**	0.66
M2V-Abundance	Accuracy	0.90 ±0.022	**0.71** ±0.032
	precision	0.90	**0.69**
	recall	0.91	**0.75**
	F1-score	0.90	**0.72**
	AUC	0.95	**0.76**

Tab. 3.6.: Classification metrics of two simulated dataset (*Null Model Nanopore* and *Ecological Nanopore*). *M2V-VR*, *M2V-MIL-VAE* and *M2V-Abundance* are our methods tested in these experiments.

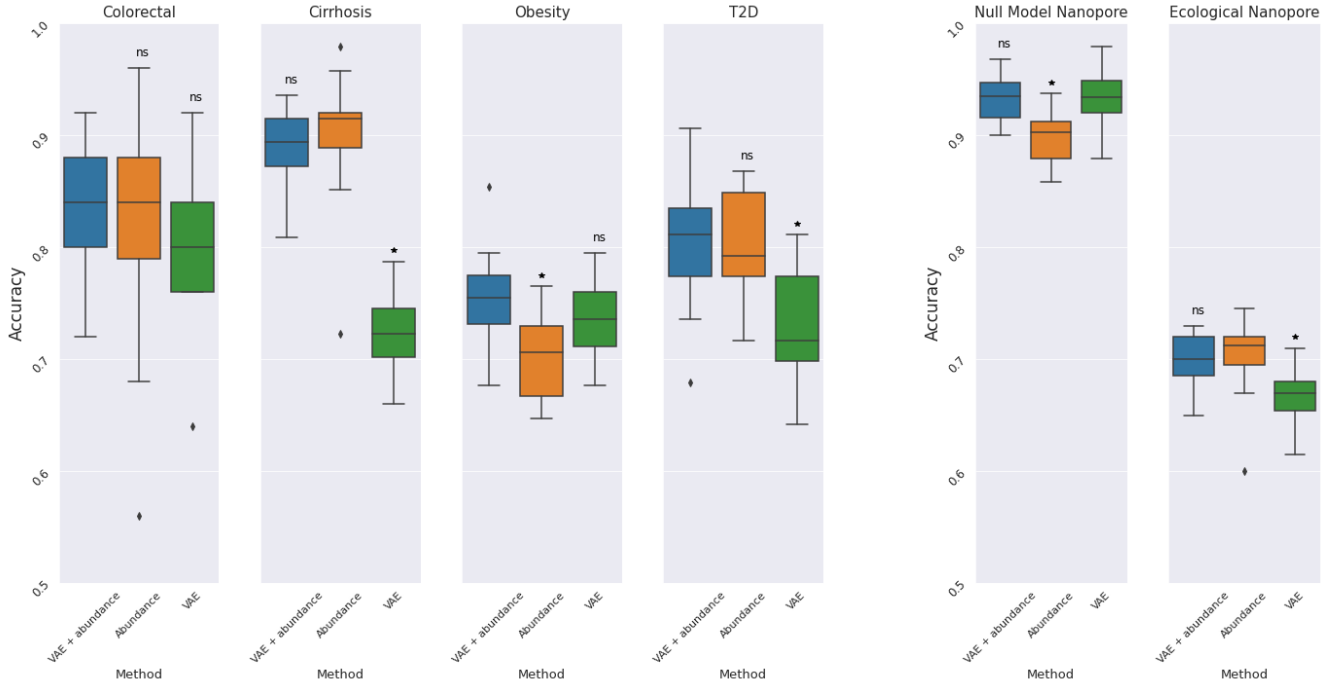
In Table 3.5, the BoK baseline obtains better results when $k = 6$ or $k = 9$ without a significant difference among them. Scores are lower than other methods, as expected, but still leads quite good results for a simple, training-free representation. Our results demonstrate that adding embeddings abstraction increases the performance and that the MIL representation yields better results.

The *M2V-MIL-VAE* architecture, compared to *M2V-MIL-DS*, leads to better results while allowing the metagenome to be represented as a single compact embeddings vector. This representation is compared to the one from *M2V-VR* and *M2V-Abundance* by visualizing their projection of the *Null Model Nanopore* dataset in a 2D space with the UMAP algorithm [MHM20] (see Figure 3.18). Nevertheless, *M2V-MIL-DS*, with an attention mechanism, adds score to genomes that could be useful to retrieve the genomes playing a role in the classification or not. We experimented this with the *Null Model Illumina* dataset. As explained in section 2.2.2, the artificial disease was

created by altering the abundance of 5 genomes compared to the abundance in the control cases. We extract, for each well-ranked positive (artificial disease) sample, the five main genomes that had the greatest impact on the outcome, resulting in a total of 12 distinct genomes. Among them, two of the five genomes invoking disease (altered abundance) were predicted at the species level at 18% and 14.5% on all well-ranked positive samples. The 12 genomes (from the top-5) are similar at the genus level with four of the genomes to be found. This shows the descriptive accuracy of the models.

Compared to our approaches, *MetaML* [Pas+16] conserves the highest precision, recall, F1-score and AUC on the Colorectal dataset but with a lower accuracy and gets lower classification scores for all metrics on the 3 other datasets. Moreover, we recall that the raw metagenomic data must be converted to an abundance table by a bioinformatics workflow before being fed to *MetaML*.

In Table 3.5 and 3.6, *M2V-Abundance* performs best in 3 of the 4 real word datasets and in 1 of the 2 simulated datasets, while *M2V-MIL VAE* gets the highest scores on the remaining datasets. It shows that the sole information of the species abundance computed by *FastDNA* is sufficient in most of the cases. We tested the combination of both representations, *M2V-MIL-VAE* and *M2V-Abundance*, to analyze the new results obtained reported in Figure 3.16a and Figure 3.16b for real-world and simulated datasets respectively.



(a) Real-World Dataset

(b) Simulated datasets

Fig. 3.16.: Box plot of the 20-fold cross validation accuracy scores. VAE, Abundance and VAE + abundance refers respectively to the 3 model representations *M2V-MIL-VAE*, *M2V-Abundance* and their combination.

Figure 3.16 illustrates the combination of the embeddings computed from *M2V-MIL-VAE* and the species abundance from *M2V-Abundance* (computed by *FastDNA*) produces results that are are comparable to the best scores among the two models. We deduce this combination is the best way using to predict metagenomic disease on the datasets used in our experiments (see Figure 3.17).

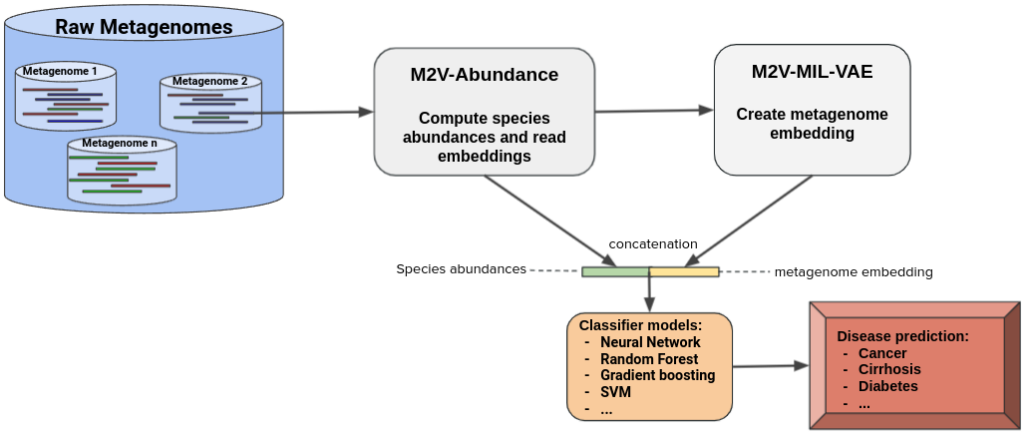
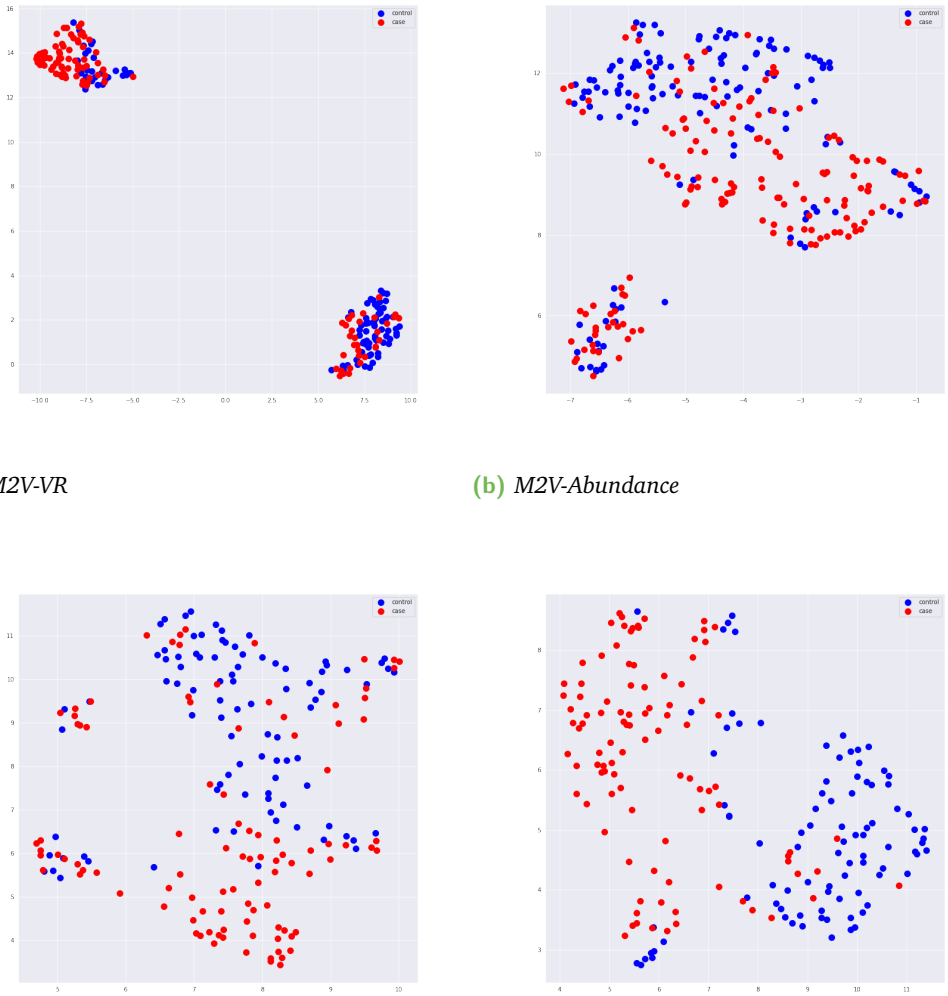


Fig. 3.17.: Our best model tested on the benchmark. It is a combination of the metagenome representations of *M2V-MIL-VAE* and *M2V-Abundance* (*FastDNA*) passed to a SOTA classifier to make disease prediction.



(a) *M2V-VR*

(b) *M2V-Abundance*

(c) *M2V-MIL-VAE without tuning*

(d) *M2V-MIL-VAE*

Fig. 3.18.: UMAP 2-D projection of the embeddings created by four different models on the *Null model Nanopore* test set.

The visualization of Figure 3.18 highlights the difference between the four embeddings representations. The vectorial representation is a drastic compression of all embeddings computed by metagenomes and create two distinct compact clusters. Control and case samples are mixed in the clusters making it difficult to classify them correctly. The projection of the species abundance representation and the *M2V-MIL-VAE* embeddings without tuning are quite similar. It forms a more dispersed point cloud with more obvious separation between classes. The Last embeddings from *M2V-MIL-VAE* tuned has the clearest separation between classes.

3.5 Conclusion

In this chapter, we discussed the use of deep learning approaches to analyze metagenomic data. Bioinformatic pipelines require huge reference catalogs of genes and important computational resources to infer the results. With deep learning, only the training stage requires such resources, while the prediction stage is based on weights learned from neural networks. This results in time savings (from several to about one hour per sample) and a reduction in the amount of data for the inference process. These features allow the diagnosis to follow a point-of-care treatment.

The developed DL model, called *Metagenome2Vec*, is composed of four steps established in this order: *Kmer2Vec* (learns k-mer embeddings), *Read2Vec* (learns read embeddings), *Read2Genome* (learns read classification) and *Metagenome2Vec* (learns metagenome embeddings and classification). To build this workflow, we have explored different architectures each one with its specificities. For *Kmer2Vec* and *Read2Vec*, we used models originally designed to compute word embeddings such as *FastText*, *word2vec* and *GloVe* or sentence embeddings such as *Sequence-to-Sequence Transformers*, that we adapted to DNA. The embeddings computed by these methods are then used with state-of-the-art classifiers for the *Read2Genome* step. We also analyzed the *FastDNA* model, an adaptation of *FastText* that gathers the three steps *Kmer2Vec*, *Read2Vec* and *Read2Genome* directly trained on the reads binning task. The *Read2Genome* part needs simulated data to train the supervised models because it gives the class information of the reads, which is not the case for reads from NGS technologies. The decomposition of our model into several stages allows us to calculate intrinsic evaluations to determine the most suitable methods. According to our analysis, *FastDNA* obtained the best performances on the intrinsic evaluations of read embeddings and on the extrinsic evaluations of reads classification. Finally, in the last step, several reads embeddings are processed with MIL models. Two MIL architectures were tested, the first one is *M2V-MIL-DS* based on *DeepSets* model to

directly predict diseases from the data with an attention mechanism allowing the model to retrieve the taxa that may be involved in the stratification. The second one is *M2V-MIL-VAE*, an adaptation of a *Variational Autoencoder* for MIL, creating a metagenome embedding as a single vector that is then passed to state-of-the-art classifiers trained to predict sample class.

Four real-world datasets, one Illumina simulated dataset and two Nanopore simulated datasets are used in our experiments. *Metagenome2Vec* achieved similar or better performances than the stat-of-the-art models such as *MetaML* or *DeepMicro*. We showed that metagenome embedding representations capture concepts relevant to the classifier to predict the class of a sample. Nevertheless, the species abundance computed at the *Read2Vec* stage (with *FastDNA*) is mainly the most important information for classification while being more understandable. Finally, after downstream experiments, we determined that our best model is the one derived from the combination of metagenome embeddings learnt by *M2V-MIL-VAE* and the *FastDNA* predicted species abundances reaching the best scores on all real-world and simulated datasets used in our experiments.

One of the main weaknesses of *Metagenome2Vec* is related to the interpretability of its predictions because our approach relies on black-box models. This is why in the next chapter we are interested in solutions based on subgroup analysis, a method capable of stratifying samples in a personalized way with understandable predictions.

4 Generate statistically credible subgroups for interpretable metagenomic signature

One of the goals of this thesis was to define an interpretable and personalized stratification method to enhance the interest of precision medicine in metagenomics. Identifying metagenomic signatures is becoming increasingly important in precision medicine. The family of subgroup discovery (SD) algorithms is particularly interesting because it meets both criteria of interpretability and personalization. To address the interpretability/accuracy trade off, we propose a hybrid approach, called *Q-Classifier*, based on a cascade classifier combining a first step of SD (for interpretability) and then a supervised model (for accuracy). With this approach, different interpretable signatures stratify the maximum possible number of patients while those remaining are defined by a default non-interpretable signature. To construct this model, we first focused on the *Q-Finder*, a SD algorithm developed by Quinten. A first step was to formalize and structure it. An overhaul of the algorithm was done to determine its place in the rule learning literature, to detail its functioning with its advantages / disadvantages and to benchmark it against the state of the art. This allowed to make a publication on the algorithm [Esn+20] and to have a better theoretical basis to adapt it into a metagenomic classifier that can be directly integrated into the pipeline described in the chapter 3. This chapter is composed first by an introduction of the SD and its application in clinical research (Section 4.1), next, the functioning of the *Q-Finder* as well as experiments and results are described (Section 4.2)[1], finally the SD application to the metagenomics with *Q-Classifier* is discussed (Section 4.3).

[1]These two sections are largely inspired by the *Q-Finder* article.

4.1 Introduction

4.1.1 Subgroup analysis in clinical research

Randomized Clinical Trials (RCTs) aim to test predefined hypotheses and answer specific questions in the context of clinical drug development. Essentially designed to demonstrate treatment efficacy and safety in a given indication using a limited number of patients with homogeneous characteristics, RCTs are performed in heavily controlled experimental conditions in order to maximize chances to obtain results with sufficient statistical power throughout successive trials. RCTs are the gold standard for evaluating treatment outcomes, although real life studies can reveal mismatches between efficacy and effectiveness [Sat+14]. Conversely, Real-World (RW) Data (electronic medical records, claims data, registries), are mainly generated for administrative purposes, going beyond what is normally collected in clinical trial programs, and represents important sources of information for healthcare decision makers.

In both RCT and RW studies, subgroup analysis (SA) is used to test local effects, for instance to account for the heterogeneity in the response to treatment. Particularly in RCT, SA "has become a fundamental step in the assessment of evidence from confirmatory (Phase III) clinical trials, where conclusions for the overall study population might not hold" [Tan+16]. SA include both confirmatory analyses, whose purpose is to confirm predefined hypotheses, and exploratory ones, which aim to generate new knowledge and are exploratory in nature [Lip+16]. When considering a set of patients included in a database, a subgroup of patients is any subset characterized by its *extension* (all the patients in the subset, e.g. Patient's ID in {"12345", "45678"}) and its *intension* (a description that characterizes the patients in the subset: e.g. "All the adult women"). In SA, a typical type of subgroups of interest are those whose extension corresponds to patients who respond differently to a new treatment [Zha+18]. A formal definition of subgroups can be found in [Lip+16].

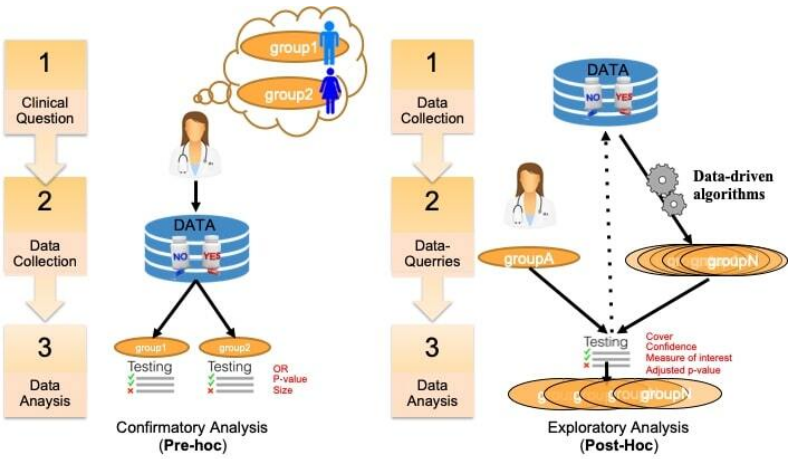


Fig. 4.1.: A classification of SA tasks distinguishing the confirmatory analyses (left) from the exploratory ones (right).

A key issue in SA in general is to assess and report its results [Rot05]. In clinical trials, this assessment is critical and depends on the precise purpose of the study. There are different ways to distinguish the purpose of using SA in clinical research. A first distinction relates to the general purpose of the analysis that can be either aimed at studying treatment efficacy or safety, on either *a priori* defined groups or *a posteriori* groups. This dichotomous classification is depicted in Figure 4.1. In the literature, pre-hoc analysis is most-often called confirmatory analysis whereas post-hoc analysis is called exploratory analysis [Lip+16].

More recently Lipkovich et al. [Lip+16] have refined this classification into four different tasks:

(A) **Confirmatory subgroup analysis:** refers to statistical analysis mainly aimed at testing a medical hypothesis under optimal setting in the absence of confounding factors while strongly controlling the type 1 error rate (using the Family-Wise Error Rate) in Phase III clinical trials with a small number of prespecified subgroups.

(B) **Exploratory subgroup evaluation:** refers to statistical analysis aimed at weakly controlling the type 1 error rate (using the False Discovery Rate) of a relatively small number of prespecified subgroups that focuses mostly on "treatment-by-covariate interactions and consistency assessments".

(C) **Post-hoc subgroup evaluation:** refers to non-data-driven statistical post-hoc assessments of the treatment effect across small sets of subgroups that include responses to regulatory inquiries, analysis of safety issues, post-marketing

activities in Phase IV trials, and assessment of heterogeneity in multi-regional studies.

(D) **Subgroup discovery:** refers to statistical methods aimed at selecting most promising subgroups with enhanced efficacy or desirable safety from a large pool of candidate subgroups. These post-hoc methods employ data mining/-machine learning algorithms to help inform the design of future trials.

We propose a decision tree to represent this second classification where the criteria to distinguish pre-hoc analysis is the strength of type 1 error control (strong or weak respectively) while for post-hoc analysis the explicit use of the collected data (hypothesis-driven or data-driven) is considered (see Figure 4.2).

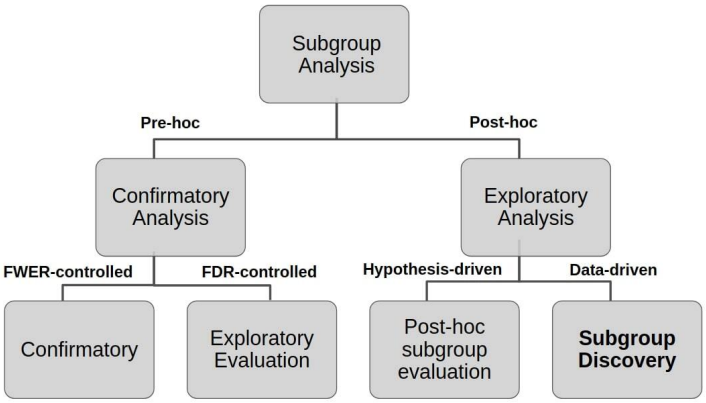


Fig. 4.2.: Hierarchical tree representing the two layers classification of SA tasks and criteria used.

The sequel of this paper is concerned with exploratory analysis that are based on Data Mining approaches and known as SD. SD has been used in a large number of applications in the medical field and data analysis of randomized clinical trials [Sun+14].

4.1.2 Subgroup discovery: two cultures

Two cultures related to subgroup discovery can be distinguished in the literature. The first one is deeply rooted in medical data analysis, biostatistics and more specifically in the context of drug discovery where both treatments arms and the outcome are key to the analysis. In this domain-specific context [Lip+18; Lip+16], that includes either or both candidate covariates and treatment-by-covariate interactions, SD algorithms search either for:

- a global modeling across the entire covariate space (e.g. Virtual Twins [FTR11], penalized logistic regression, FindIt [IR13], Interaction Trees [Su+09] which extends CART to include treatment-by-covariate interactions, ...).
- a local modeling that focuses on identifying specific regions with desirable characteristic (e.g. SIDES [LD14], PRIM [PW10], TSDT [BSR14], ...).

- a global modeling across the entire covariate space (e.g. Virtual Twins [FTR11], penalized logistic regression, FindIt [IR13], Interaction Trees [Su+09] which extends CART to include treatment-by-covariate interactions, ...).
- a local modeling that focuses on identifying specific regions with desirable characteristic (e.g. SIDES [LD14], PRIM [PW10], TSDT [BSR14], ...).

The second culture of SD is rooted in the Data Mining and KDD community and applies to any kind of data. The related fields include association rules, set mining, contrast sets, emerging patterns all relating to the notion of descriptive induction [FGL12].

Although both cultures share common requirements and issues, their vocabulary differs and are practically mutually exclusive in the SD literature. We propose a hierarchical tree representing both cultures and their main associated algorithms (see Figure 4.3). Since the *Q-Finder* approach we propose in this paper inherits from both cultures, it is worthwhile giving an account of both of them.

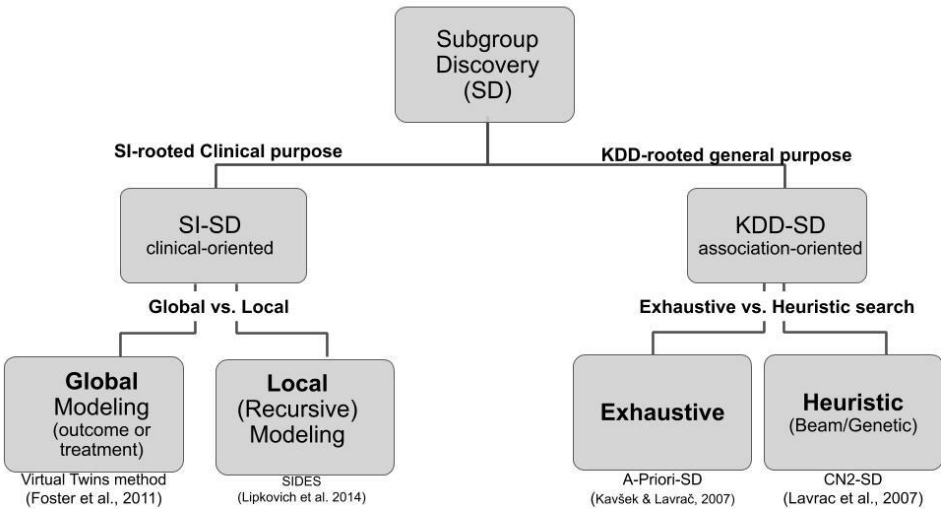


Fig. 4.3.: Hierarchical tree representing the SD approaches in both biomedical data analysis and data mining cultures. The references under the boxes correspond to representative algorithms of each kind.

In the first culture, where SD is also often referred to as SI [Bal+18; Che+17; DL14; HY18; Lip+16; Lip+17; Xu+15; Zha+18], there is a key distinction between prognostic factors (supporting identification of patients with a good or poor outcome regardless of the treatment assignment) and predictive factors (supporting identification of patients' response to the treatment) [AS00].

In this culture, SD algorithms[2] can be distinguished depending on whether they search for prognostic and/or predictive factors: the ones that can only look for predictive factors (Quint [DDM16], SIDES, Virtual Twins, Interaction trees, ...), the ones that only look for prognostic factors (PRIM, CART [HKH18], ...), and the ones that can look for both prognostic and predictive factors (STIMA [DCV10], MOB [ZHH08], ...). The key measures to assess the quality of the SD results in this culture are p-value, type 1 errors, False-Discovery Rate [Lip+18; Lip+16].

In the second culture, SD is not associated with a specific sector such as clinical research. On the contrary, SD is defined as "given a population of individuals and a property of those individuals that we are interested in, [the finding of] population subgroups that are statistically the 'most interesting', e.g., are as large as possible and have the most unusual statistical (distributional) characteristics with respect to the property of interest" [FGL12]. More generally, SD "is a type of data mining technique that supports the identification of interesting and comprehensible associations in databases, confirming hypotheses and exploring new ones" [Atz15]. These associations are in the form of a set of rules represented as Subgroup $\rightarrow$ Target, where Target is the property of interest (e.g. $Hypoglycemia = Yes$) and Subgroup is a conjunction of attribute-selector-value triplets (e.g. $Age > 18$ & $Sex = F$). SD belongs to the wider domain of Association Rule mining — this explains why many algorithms bear a name formed from an association rule algorithm and an SD extension — and differs from classical supervised learning as the goal is not to find rules that best predict the target value of unknown observations but rather best support describing groups of observations that when satisfying the condition of a rule also satisfy the target [FGL12].

In this second culture the SD process consists in three main phases: candidate subgroup generation, subgroups evaluation and ranking [Hel16], and subgroups pruning (e.g. top-k pruning). The key issues being more related to the algorithmic search for subgroups than their evaluation. This includes the search strategy (be it beam [SD, CN2-SD, Double-Beam-SD], exhaustive [APRIORI-SD, Merge-SD] or genetic [SD-IGA, SGBA-SD]), stopping criterion (minsup, minconf, maxsteps, etc.)

[2] We focus here on subgroup discovery algorithms which, unlike classification algorithms, meet the objective of discovering interesting population subgroups rather than maximizing the accuracy of the classification of the induced set of rules [Lav+04].

[Val+17a], pruning technique (constraint, minimum support or coverage) and quality measures (confidence, support, usualness [CN2-SD, APRIORI-SD], etc.).

Recent theoretical and empirical analyses have elucidated different types of methods to select algorithms suitable for specific domains of application [Hel16]. Applying such algorithms to SA requires considering the outcome as the variable of interest. Nevertheless, the treatment is not explicitly considered as a special variable and dozens of quality measures exist (number of rules, number of variables, support, confidence, precision, interest, novelty, significance, false positive, specificity, unusualness (WRAcc), etc.) [Her+10].

We will refer to Subgroup Discovery in the context of clinical Subgroup Identification as SI-SD and to Subgroup Discovery in the context of Knowledge Discovery in Database as KDD-SD and compare them with the *Q-Finder* approach. There is an extensive literature comparing algorithms belonging to each culture independently (e.g. [Doo+13; Zha+18; LCZ19]) but, to our knowledge, they are not compared when they come from two different cultures.

4.1.3 Limits of current SD algorithms for clinical research

4.1.3.1. Lack of statistical power and hypothesis generation

As stated by Burke et al. [Bur+15] "the limitations of subgroup analysis are well established —false positives due to multiple comparisons, false negatives due to inadequate power, and limited ability to inform individual treatment decisions because patients have multiple characteristics that vary simultaneously". Controlling such errors is a problem: a survey on clinical industry practices and challenges in SD quoted the lack of statistical power to test multiple subgroups as a major challenge [MLD15]. Consequently, SI-SD algorithms often fail to detect any "statistically significant" subgroups.

To control for multiple testing errors SI-SD algorithms often rely on approaches that drastically restrict the number of explored candidate subgroups at the expense of hypotheses generation, usually by using recursive partitioning [Doo+13]. Recursive partitioning approaches could miss emerging synergistic effects, defined as subgroups associated to the outcome, whose individual effects (related to each attribute-selector-value triplet) are independent from the outcome [HHZ10]. As such, individual effects combinations would not be selected in tree nodes. Equally, recursive partitioning may also miss optimal combinations of attribute-selector-value

triplets, as an optimal selector-value for a given attribute is only defined with relation to previously defined attribute-selector-value triplets[3] [HHZ10]. Therefore, subgroups in output are defined by a combination of variables for which thresholds are not necessarily the optimal ones (with respect to the metrics of interest to be optimized). Furthermore, search space restriction strategies favor the detection of the strongest signals in the dataset, that are often already known and/or redundant from each other

Finally, pure beam search strategies could miss relevant subgroups as they try to optimize the joint, i.e. global, accuracy of all leaves, that is a tree with the most heterogeneous leaves. Consequently, when limiting the complexity (i.e. subgroups length), we can miss interesting local structures in favor of the global picture[4] (see section A.1.1 in supplementary materials that shows an example where beam search strategy using a decision tree misses relevant subgroups).

On the contrary, KDD-SD approaches support the exploration of much wider search spaces at the expense of accuracy, as they do not in general control for type 1 errors (be it strong or weak).

4.1.3.2. Insufficient credibility and acceptance of subgroups

The "Achille's heel" of SD is the question of credibility of its results. Several meta-analyses have demonstrated that discovered subgroups rarely lead to expected results and have proposed criteria to assess the credibility of findings [Rot05]. Such credibility metrics are key to support confidence in subgroups and their acceptance by regulatory agencies and publication journals. Several credibility metrics have been provided and recommended [Rot05; Sun+10; Dij+09] such as the type of measures of association (relative risk, odds-ratio, ...), correction for confounders, correction for multiple testing, as well as treatment-covariate interaction tests.

SI-SD approaches use credibility metrics suited to clinical analyses. However, most of them only provide and consider in their exploration a limited number of credibility metrics (e.g. hypothesis testing p-value), compared to what is recommended in the literature. Moreover, such metrics are rarely consensual. Equally, the subgroups'

[3] Let's assume that a recursive partitioning algorithm has defined $BMI > 25$ as the optimal attribute-selector-value triplet on an objective function to be optimized for patients with $Age > 18$ (the latter being the first triplet to be identified by the algorithm). One can assume that better selector-values could have been obtained for this combination of attributes, to generate the optimal combination of these attributes on the objective function (e.g. $Age > 21$ & $BMI > 20$).

[4] Further explanation here: http://www.realkd.org/subgroup-discovery/the-power-of-saying-i-dont-know-an-introduction-to-subgroup-discovery-and-local-modeling/

generation process (that defines optimal attribute-selector-value triplets combination) mostly relies on the optimization of a limited number of criteria and is thus not directly driven by all credibility metrics that will be used for the clinical assessment of the subgroups at the end.

On the other hand, KDD-SD can provide a considerable range of credibility metrics as there is no consensus about which quality measures to use [Her+10], such as WRAcc, Lift, Conviction, Mutual information [Hah+11]. However, these metrics are seldom used in clinical analyses, hindering their use in the medical field.

Another issue hindering the adoption of SD approaches lies in the comprehensibility of the algorithm itself. This often-underestimated issue is an obstacle for convincing clinical teams and regulatory agencies of the relevance and reliability of SD approaches.

4.2 *Q-Finder*'s pipeline to increase credible findings generation

In this section we present an approach that aims at combining some of the advantages of both SI-SD and KDD-SD cultures, while dealing with limitations observed in current SD algorithms (see section 4.1.3). To this end, we introduce *Q-Finder*, which relies on a four-steps approach (summarized in Figure 4.4): exhaustive subgroup candidates generation, candidate subgroups assessment on a set of credibility metrics, selection of a limited number of most promising subgroups that are then tested during the final step. This approach has been applied in several therapeutic areas, with published examples available [Alv+20; Mor+20; Iba+19; Zho+19; Zho+18; Rol+18; Dum+18; Gas+17; Dum+16; Ada+16; Amr+15; Eve+14; Nab+12].

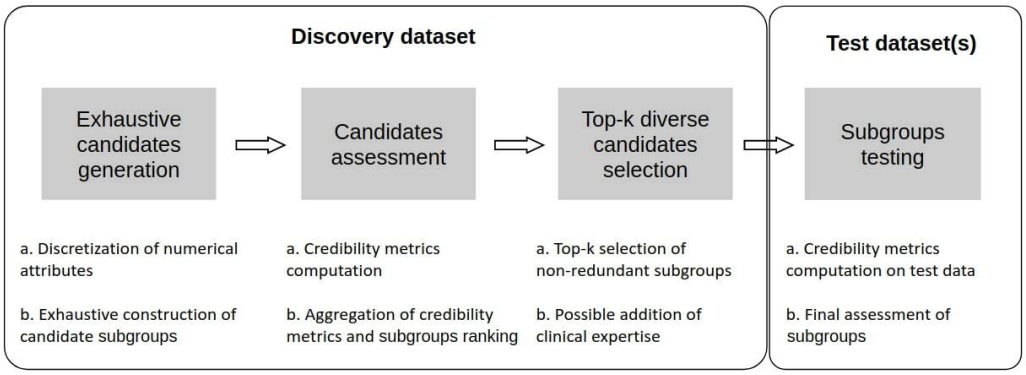


Fig. 4.4.: *Q-Finder* works in 4 main stages: an exhaustive generation of candidate subgroups, a ranking of candidate subgroups via an evaluation of their empirical credibility, a selection of the best candidates (taking into account the redundancy between subgroups) then an assessment of subgroups' credibility on one or more test datasets

4.2.1 Basic definitions: patterns, predictive and prognostic rules

Numerous formalizations of KDD-SD have been given in the literature. We will briefly introduce some basic definitions of database, individuals, basic patterns, complex patterns, subgroup complexity and subgroup description related to the ones introduced by Atzmueller [Atz15]. A database is formally defined as $D = (I, A)$, a set I of N individuals and a set A of K attributes. We will only distinguish nominal and numerical attributes. For nominal attributes, a basic pattern $(a_i = v_{i,j})$ is a Boolean function that is true if the value of attribute $a_i \in A$ is equal to $v_{i,j}$ in the domain of a_i for a given individual of I. For a numerical attribute (be it real or integer) a_i, both basic patterns $(a_i \geq v_{i,j})$ and $(a_i \leq v_{i,j})$ can be defined for each value $v_{i,j}$ in the domain of a_i. The associated Boolean function is defined similarly. The set of all basic patterns is denoted by Σ.

A conjunctive language is classically considered to describe subgroups. An association rule $(X \rightarrow Y)$ is composed of a complex pattern (also called itemset) X and a basic pattern Y, where X is called antecedent (or left-hand-side (LHS) or Subgroup) and Y the consequent (or right-hand-side (RHS) or Target). A complex pattern CP is described by a set of basic patterns $CP = \{BP_1, \ldots, BP_k, \ldots BP_C\}, BP_k \in \Sigma$. It is logically interpreted as a conjunction of basic patterns. In other words, a complex pattern CP represents the body of a rule $BP_1 \wedge ... \wedge BP_C$. In *Q-Finder*, its length C corresponds to the *complexity* of the associated rule. The set of observations covered by a complex pattern CP is called the *extension* of the subgroup, i.e. the individuals for which CP is true $\{x \in I$; CP is true for x$\}$. In this formalism, the

set of all possible association rules is included in the powerset of Σ although many subsets are not considered because their extension is by construction empty (e.g. $a_i \geq v_{i,j} \wedge a_i \leq v_{i,k}$ when $v_{i,j} > v_{i,k}$). Moreover, this set of all subgroups can be *partially ordered* in a lattice structure [Gan93]. We will not rely on such lattice structure because the length of subgroups (i.e. their complexity) is sufficient to partially order the set of generated candidates in subsets[5].

In SI-SD, many databases include information about treatment distinguishing different individuals grouped in arms. This notion is critical to distinguish two types of rules. The prognostic rules are not related to a treatment effect on a given outcome, unlike the predictive rules.

These two main types of rules can be summarized as follows:

Prognostic rule: SUBGROUP $\rightarrow$ TARGET

Predictive rule: SUBGROUP where TREATMENT $\rightarrow$ TARGET

4.2.2 Preprocessing and Candidate Subgroups generation in *Q-Finder*

In *Q-Finder*, to control the size of the set of basic patterns $|\Sigma|$, all numerical attributes are systematically discretized in bins. A hyperparameter $\#Bins$ sets the maximum number of values $v_{i,j}$ of any numeric attribute a_i (default value: 10). If this number is above $\#Bins$, the attribute a_i is quantized using a discretization method $DiscretizationMethod$ (see algorithm 25 line 8). Different methods exist for quantization, the default one being equal-frequency binning. In the same way, the number of distinct values for a given nominal attribute might be bounded by the hyperparameter $\#Cats$ (default value[6]: ∞). If the number of modalities is above this threshold, a reduction method $ReductionMethod$ may be used (by default: use the $(\#Cats - 1)$ most frequent values of a_i and a create a value "other" for all the remaining ones). Let us call Kc the number of nominal attributes and Kb the number of numerical attributes. After this preprocessing step the number of basic patterns $|\Sigma|$ is bounded and we have the relation: $|\Sigma| \leq 2 * Kb * \#Bins + Kc * \#Cats$.

Given a set of basic patterns Σ, we call "candidate generation" the search procedure that generates the subgroups (i.e. complex patterns conjunction of basic ones).

[5] A methodology to further order the subgroups is introduced in section 4.2.3.2
[6] In this way, no reduction is done by default.

The number of complex patterns of complexity C is bounded by the number of C-combination of Σ (i.e. the binomial coefficient $\binom{|\Sigma|}{C}$). There is extensive literature in KDD-SD on the type of exploration of these complex patterns [FGL12]. Experiments have shown that the exhaustive search-based methods perform better than other methods which prune the search before evaluation [Hel16]. This is particularly true when the problem size is reasonable (i.e. a few thousand individuals) which is mostly the case in SD. The *Q-Finder* candidate generation is straightforward; it outputs a subset of all *C-combinations* of Σ (with $C \in [\![1; C_{max}]\!]$) as described below in Algorithm 25.

Algorithm 1: Basic patterns and candidate subgroup generation of complexity $\leq C_{max}$

input : $\boldsymbol{\#Bins}$: sets the maximum number of values $\boldsymbol{v_{i,j}}$ of any numeric attribute
$\boldsymbol{\#Cats}$: Bounds the number of distinct values for a given nominal attribute
$\boldsymbol{C_{max}}$: maximum complexity of generated subgroups
$\boldsymbol{ReductionMethod}$: by default, uses the $(\boldsymbol{\#Cats - 1})$ most frequent values of $\boldsymbol{a_i}$ and creates a value "other" for all the remaining ones
$\boldsymbol{DiscretizationMethod}$: Method to quantize $\boldsymbol{a_i}$

output : $\boldsymbol{G}$ the set of generated candidate subgroups of length $\leq \boldsymbol{C_{max}}$

```
// Set of basic patterns
Σ = {}
for each nominal attribute a_i do
    if #valueof(a_i) > #Cats then
        Reduce the number of values of a_i to #Cats using ReductionMethod
    end
end
for each v_{i,j} do
    Σ = Σ ∪ {(a_i = v_{i,j})}
end
for each numerical attribute a_i do
    if #valueof(a_i) > #Bins then
        Discretize the values of a_i in #Bins using DiscretizationMethod
    end
end
for each v_{i,j} do
    Σ = Σ ∪ {(a_i ≥ v_{i,j}), (a_i ≤ v_{i,j})}
end
// # Set of generated subgroups
G = {}
for each combination s of 1 to C_max elements of Σ do
    if one attribute a_i appears twice or more in s or if the extension of s is empty by construction then
        skip
    else
        G = G ∪ {s}
    end
end
```

In practice the *Q-Finder* algorithm not only supports constructing left-bounded and right-bounded intervals but also supports bounded intervals depending on the

number of basic patterns (one or two) associated to a given numerical attribute. If bounded intervals are considered, step 13 of the algorithm becomes "**If** one attribute a_i appears twice or more in s with the same selector or if the extension of s is empty by construction **then skip**".

4.2.3 Empirical credibility of subgroups

Q-Finder's candidates generation step may potentially produce a very large number of subgroups. Because of its exhaustive strategy, it produces a number of subgroups which grows exponentially with complexity. Dealing with a massive exploration of database is the challenge of any KDD-SD algorithm be it exhaustive or heuristic, as the number of computed statistical tests may induce a high risk of false positives, that needs to be mitigated.

Q-Finder addresses this challenge by only selecting a subset of candidate subgroups and testing them on independent data, to assess the replicability of the results while controlling the number of tests (and thus the type 1 error). This strategy requires to address two issues:

- a way of evaluating the empirical credibility of subgroups, in order to rank them from most to least promising
- a top-*k* selection strategy, in order to select a set of subgroups that seem most credible and will be tested on an independent dataset.

4.2.3.1. Credibility metrics

The notion of credibility often appears in the literature on subgroup analysis [Bur+15; Sch+16; Sun+10; SBJ12; Dij+09] described according to different criteria. In particular Oxman and Guyatt [OG92] detail seven existing criteria to help clinicians assess the credibility of putative subgroup effects on a continuum from "highly plausible" to "extremely unlikely". Sun et al. [Sun+10] suggest four additional credibility criteria and re-structure a checklist of items addressing study design, analysis, and context. In the present context, credibility is related to a sequence of *a priori* ordered statistical metrics that are progressively increasing the confidence (credibility) of a given subgroup. The seven criteria described below are aligned with the clinical domain endpoints [Sun+10; Dij+09]. Using these criteria when selecting the top-ranked subgroups ought to both promote the finding

of credible subgroups and facilitate their acceptance by clinicians, agencies and publication journals.

Drawing from this literature, continuous metrics to measure subgroups' credibility are used in *Q-Finder*.). Several *credibility criteria* are defined, each composed of both a continuous metric and a minimum or maximum threshold (which may be modified by the user):

1. **Coverage criterion:** The coverage metric is defined by the ratio between the subgroup's size and the dataset's size. This allows to only consider the subgroups that correspond to large enough groups of patients to be clinically relevant. It can be compared to defining a minimum SUPPORT of the antecedent of a rule in the KDD-SD literature. Default minimum threshold for coverage is 10%.

2. **Effect size criterion:** As recommended by both Sun et al. [Sun+10] and Dijkman et al. [Dij+09], *Q-Finder*'s exploration relies by default on relative risk reductions, which differ according to the probability distribution of the outcome (ODDS-RATIOS for discrete or negative binomial distributions, RISK-RATIOS for normal or Poisson distributions, HAZARD RATIOS for survival analysis). Those metrics allow to quantify the strength of the association between the antecedent (the subgroup) and consequent (the target) of the rule. Relative risk reductions remain, in most situations, constant across varying baseline risks, in comparison to absolute risk reductions. In the KDD-SD literature, this continuous metric is usually the CONFIDENCE (i.e. how often the target is true among the individuals that satisfy the subgroups).

   The effect size metric may vary depending on whether one is looking for predictive or prognostic factors. When searching for prognostic factors, *Q-Finder* only considers the effect size measuring the subgroup's effect (default minimum threshold for effect size is 1.2). When searching for predictive factors, *Q-Finder* considers simultaneously two effect sizes: the *treatment effect within the subgroup* and the *differential treatment effect*, defined as the difference in treatment effect for patients inside the subgroup versus outside the subgroup. When generating predictive factors, one can consider the *differential treatment effect* on its own, or in combination with the *treatment effect within the subgroup*. The latter case allows to identify subgroups in which the treatment effect is both positive and stronger than outside the subgroup (default thresholds are 1.0 for the *treatment effect within the subgroup* and 1.2 for the *differential treatment effect*).

3. **Effect significance criterion:** the association between each subgroup and the target is assessed using a nullity test from a generalized linear model. For the identification of predictive factors, an interaction test is performed to assess between-subgroup treatment effect interactions as recommended by Dijkman et al. [Dij+09]. A threshold (typically 5%) is used to define when the p-value related to each effect size metric is considered significant.

4. **Basic patterns contributions criteria:** Basic patterns contributions to the subgroup's global effect are evaluated through two sub-criteria: the absolute contribution of each basic pattern and the contributions ratio between basic patterns.

   The *absolute contribution* of a basic pattern is defined by the improvement in effect when this basic pattern is present, compared to the subgroup's effect when this basic pattern is absent. Each basic pattern contribution should be above a defined threshold (by default 0.2, 0 and 0.2 respectively for the subgroup's effect, the *treatment effect within the subgroup* and the *differential treatment effect*), thus ensuring that each increase in subgroup's complexity goes along with some gain in effect and therefore in interest.

   The *contributions ratio* between basic patterns is the ratio between the maximum *absolute contribution* and the minimum *absolute contribution*. A maximum threshold (by default 5 for the subgroup's effect or the *differential treatment effect*) is set for this criterion, thus ensuring that basic patterns' contributions to the subgroup's effect are not too unbalanced. Indeed, if a basic pattern bears only a small portion of the global subgroup's effect, then the global effect's increase is not worth the complexity's increase due to this pattern's addition.

5. **Effect size criterion corrected for confounders:** the strength of the association is assessed through relative risk reductions (as in criterion 2) while correcting for confounding factors using a generalized linear model. Added covariates are known confounding factors of the outcome, which are susceptible to be unbalanced between patients within and without each subgroup, as well as between treatment arms for predictive factors identification tasks [Sun+10; Dij+09]. As for criterion 2, adjusted relative risks ought to be above a given threshold (same as for criterion 2).

6. **Effect significance criterion corrected for confounders:** as for the effect significance criterion (criterion 3) and using the same model as in criterion 5,

a threshold (typically 5%) is used to define when the p-value related to each effect size metric corrected for confounders is considered significant.

7. **Effect adjusted significance criterion corrected for confounders:** the p-value computed in criterion 6 is adjusted to account for multiple testing, as recommended by Dijkman et al. [Dij+09]. This procedure relies on a Bonferroni or a Benjamini-Hochberg correction to control for type 1 errors. As for criterion 6, a threshold is used to determine whether the p-value remains significant after multiple testing correction (typically 5%)

These seven credibility metrics are at the core of *Q-Finder*. However, they can be further extended by other measures of interest to better fit each research question.

4.2.3.2. Aggregation rules and subgroups ranking

Aggregation rules are defined to discriminate subgroups according to a set of criteria and therefore to help select the most interesting and/or promising ones for each research question. This is a key concept of *Q-Finder*, as the goal is to select a set of "top" subgroups before testing them on an independent dataset, whether they pass all credibility criteria. In practice, ranking subgroups into aggregation ranks is helpful when no subgroup passes all credibility criteria, and we need to investigate lower aggregation ranks to select the most promising subgroups. This approach contrasts with most SI-SD algorithms, where outputs are only subgroups passing all predefined indicators, hindering the generation of hypotheses if these are difficult to achieve.

To this end, a set of credibility criteria is parameterized by the user, depending on the desired properties of the searched subgroups (see section 4.2.3.1). *Q-Finder* computes each metric for each of the candidate subgroups of complexity $C \leq C_{max}$ and verifies if the associated thresholds are met. A vector of Boolean can thus be associated to each subgroup depending on which thresholds are met, and are used to order the candidate subgroups, according to prespecified aggregation rules.

By default, *Q-Finder* prioritizes subgroups that meet the following credibility criteria: subgroups with a minimal value of coverage (**coverage criterion**), defined by basic patterns that sufficiently contribute to the subgroup's effect (**basic patterns contribution criteria**), with a minimal level of effect size adjusted for confounding factors[7] (**effect size criterion corrected for confounders**) and adjusted p-values

[7] Looking for subgroups with a predefined minimal effect size is aligned with recent recommendations from the American Statistical Association [WSL19]: "Thoughtful research includes careful consideration of the definition of a meaningful effect size. As a researcher you should communicate this

for multiple testing below a given level of risk (**effect adjusted significance criterion corrected for confounders**). Please note that the above-mentioned effect could either be the subgroup's effect size (for prognostic factors) or the *treatment effect within the subgroup* and/or the *differential treatment effect* (for predictive factors). Aggregation rules are the following (from least to most stringent):

- Rank 1: subgroups that satisfy the coverage criterion
- Rank 2: subgroups of rank 1 that also satisfy the effect size criterion
- Rank 3: subgroups of rank 2 that also satisfy the basic patterns contribution criteria
- Rank 4: subgroups of rank 3 that also satisfy the effect significance criterion
- Rank 5: subgroups of rank 3 or 4 that also satisfy the effect criterion corrected for confounders
- Rank 6: subgroups of rank 5 that also satisfy the effect significance criterion corrected for confounders
- Rank 7: subgroups of rank 6 that also satisfy the effect adjusted significance criterion corrected for confounders

One can notice that subgroups with an odds-ratio adjusted for confounders but not significant (rank 5) are ranked before subgroups with significant odds-ratios (not adjusted for confounders, rank 4) for hypotheses generation. This ranking is consistent with favoring adjusted odds-ratios with a lack of statistical power to potential biased estimates. As well as the possibility of adjusting the list of parameters, the order of priority between parameters can also be changed to take into account different priorities.

In addition, a continuous criterion is chosen to sort subgroups of the same aggregation rank. Classically, the criterion called *Effect significance criterion corrected for confounders* is preferred. This is consistent with recommendations by Sun et al. [Sun+10] that state that the smaller the p-value, the more credible the subgroup becomes. In case of a tie, additional criteria can be used to determine the final

up front, before data are collected and analyzed. Then it is just too late as it is easy to justify the observed results after the fact and to over-interpret trivial effect sizes as significant. Many authors in this special issue argue that consideration of the effect size and its 'scientific meaningfulness' is essential for reliable inference (e.g., [Blu+18]; [Bet19])."

ranking, such as the *effect size criterion corrected for confounders*, to favor subgroups with stronger effect sizes. This ranking procedure is summarized in algorithm 15.

Algorithm 2: Ranking candidate subgroups

input : $\boldsymbol{G}$: the list of candidate subgroups of length $\leq \boldsymbol{C_{max}}$
$\boldsymbol{m_c}$: a continuous credibility metric (*e.g.* a p-value)
$\boldsymbol{M}$: the list of credibility criteria (*e.g.* $[(p\text{-}value < \mathbf{5\%}), (\boldsymbol{OR} > \mathbf{1})]$)
$\boldsymbol{AggregrationRules}$: Set of criteria to discriminate subgroups

output : $\boldsymbol{G_{ranked}}$: The list of subgroups of $\boldsymbol{G}$ sorted according to $\boldsymbol{Ranks}$

```
// Sort G according to m_c
G_sorted = sort(G, m_c)
// Create a vector of |G| zeros to store ranks of each s_i ∈ G
Ranks = rep(0, |G|)
for s_i in G do
    // vector representing the subgroup's credibility
    cred = [ ]
    for m_j in M do
        if s_i passes credibility criteria m_j then
            cred[j] = 1
        else
            cred[j] = 0
        end
    end
end
/* Integer part of the rank of s_i is the aggregation rank given by
   AggregationRules applied to cred                                  */
⌊Ranks[i]⌋ = AggregationRules(cred)
// Fractional part of the rank of s_i is the index of s_i in G_sorted
{Ranks[i]} = index(s_i, G_sorted)
G_ranked := sort(G, Ranks)
```

4.2.4 *Q-Finder* subgroups diversity and top-*k* selection

4.2.4.1. Subgroups diversity

Q-Finder performs a subgroups top-*k* selection to be tested on an independent dataset. One of the known issues in KDD-SD of top-*k* mining algorithms is that they are prone to output redundant subgroups as each subgroup is considered

individually. Several authors including Leeuwen and Knobbe [LK12] have argued to search for subgroups that offer a high diversity: diverse subgroup set discovery. Therefore, the goal is to take into account the fact that many subgroups might be redundant either extensionally (their basic patterns are very similar) or intentionally (the objects covered by the subgroup are similar). A general approach to address this issue is to define a redundancy measure. It can for example consider the number of common attributes between two subgroups, or the percentage of common examples covered by two different subgroups. The last requires more computation but results in a better diversification of subgroups as it considers possible correlations between variables.

Q-Finder proposes a definition of intentional redundancy between basic patterns, where two basic patterns (attribute-selector-value triplets, respectively $a_1 - s_1 - v_1$ and $a_2 - s_2 - v_2$) are considered redundant if:

- $a_1 = a_2$
- AND:
  - For nominal attributes: $v_1 = v_2$
  - For numerical attributes:
    - $s_1 = s_2$
    - OR considering s_1 as "$\leq$" and s_2 as "$\geq$", $v_1 \geq v_2$

Based on the basic patterns redundancy definition, two subgroups are called redundant if C_{min} basic patterns are redundant between them; C_{min} being the minimum complexity of the two subgroups.

4.2.4.2. Selection of top-*k* subgroups to be tested

Different strategies exist to identify an optimal top-*k* selection of non-redundant subgroups [XBM06], based on subgroups' intensions, extensions, or both. In addition to those existing strategies, *Q-Finder* proposes its own approach based on subgroups' intensions (see Algorithm 30) to determine an optimal set of k non-redundant subgroups S_k from the ranked set of generated subgroups G_{ranked} (output from Algorithm 15).

The best candidate subgroup is iteratively selected using 2 continuous metrics: m_c from Algorithm 15 and another continuous metric. This top-*k* algorithm was

originally designed using a p-value metric[8] for m_c and an effect size[9] for the second metric[10]. For the sake of clarity, we will describe this algorithm using those 2 metrics:

- Subgroups should be selected from less complex to most complex (favoring less complex subgroups)
- When two subgroups of equal complexity are redundant, only the one associated with the best p-value should be retained.
- When two subgroups of different complexities are redundant
  - The most complex subgroup of the two is discarded iff its chosen effect size metric is lower than the less complex one.
  - The less complex subgroup of the two is discarded iff both its chosen p-value and effect size metric are respectively higher and lower than the more complex one[11]

[8] P-value credibility metric can be chosen from metrics 3, 6 or 7 presented in 4.2.3.1
[9] Effect size credibility metric can be chosen from metrics 2 or 5 presented in 4.2.3.1
[10] The user can adapt this algorithm using any relevant continuous metrics' couple
[11] Note that instead of discarding the less complex subgroup of the two, one might want to keep both. The algorithm will need to be revised accordingly.

This top-k selection process based on these principles is detailed in Algorithm 30.

Algorithm 3: *Q-Finder*'s iterative top-*k* selection based on subgroups' intensions

input : $\boldsymbol{k}$: maximum number of selected subgroups
$\boldsymbol{G_{ranked}}$: set of ranked generated subgroups, with complexities ranging from $\boldsymbol{C_{min}}$ to $\boldsymbol{C_{max}}$
$\boldsymbol{\delta_{ES}}$: minimum delta to consider that a subgroup has a higher effect size[12]
output : $\boldsymbol{S_k}$: top-k best candidate subgroups

```
/* split G_ranked by subgroup complexity (G_split[1] corresponds to complexity 1,
G_split[2] to complexity 2, ...)                                              */
1  G_split = splitByComplexity(G_ranked)
   // Initialize S_k, the set of top candidate subgroups
2  S_k = {}
3  for c = C_min to C_max do
       // g :  candidate subgroup
4      for g in G_split[c] do
5          if p-value(g) > max(p-values(S_k) and size(S_k) == k then
6              continue to next c
7          end
           // s :  subgroup in the top-k
8          for s in S_k do
9              if redundant(g, s) then
10                 if complexity(g) == complexity(s) then
11                     continue to next g
12                 end
13                 if complexity(g) > complexity(s) then
14                     if EffectSize(g) ≤ EffectSize(s) + δ_ES then
15                         continue to next g
16                     end
17                 end
18             end
19         end
20         for s in S_k do
21             if redundant(g, s) and complexity(g) > complexity(s) and EffectSize(g) > EffectSize(s) + δ_ES
                 and p-value(g) < p-value(s) then
22                 S_k = S_k\{s}
23             end
24         end
25         S_k = S_k ∪ {g}
26         while size(S_k) > k do
27             S_k = S_k\{subgroup from S_k with the highest p-value}
28         end
29     end
30 end
```

The result of this step is a set of most promising non-redundant subgroups, that has a maximum size of k.

[12] Above that delta value, the increase in effect size is worth enough to justify an increase in complexity.

4.2.5 Possible addition of clinical expertise

Clinical input can be used to overrule algorithm's preference during top-k selection, by removing candidate subgroups from G_{ranked} (the set of candidate subgroups cf. Algorithm 30) or force the addition of a subgroup into S_k (the set of best candidates cf. Algorithm 30). More generally, clinical experts can directly select top-k relevant subgroups among the most credible ones. This stage, that is sometimes referred to as *Interactive Machine Learning* [Hol16], is aligned with the American Statistical Association recommendations that encourage researchers for seeking experts judgement in any statistical analysis, including for evaluating the importance and the strength of empirical evidence [WSL19]. By integrating experts into *Q-Finder*'s process for subgroups selection, one allows the consideration of non-measurable properties, such as the novelty, interest or applicability of the proposed subgroups[13].

4.2.6 Subgroups' generalization credibility

In *Q-Finder* the final step consists in computing the credibility metrics of the top-k subgroups on the testing set, in order to assess their generalization credibility, that is subgroups consistency across databases [Sun+10; Dij+09]. However, contrary to the candidate subgroups generation phase previously performed, the number of tested subgroups in this phase is well-controlled (as recommended in Sun et al. [Sun+10] and Dijkman et al. [Dij+09]), as it is limited by the parameter k. This allows a better control of the type 1 error that was more difficult to achieve until then. For that purpose, *Q-Finder* performs a correction for multiple testing during computation of the significance metrics, to account for the number of subgroups tested on independent data (default: Benjamini-Hochberg procedure). top-*k* subgroups satisfying the credibility criteria on the test dataset are considered highly credible.

4.2.7 Experiments and Results

This section is dedicated to compare *Q-Finder* with representative algorithms for predictive or prognostic SD. First, the IDMPS database on which experiments were run is described. Then, the research questions are stated and both a prognostic and

[13] Wasserstein et al. [WSL19] argue to be open in study designs and analyses: "One might say that subjectivity is not a problem; it is part of the solution."

a predictive task are described. Lastly, four different methods and their results are given and compared with *Q-Finder*.

4.2.7.1. Research questions

Prognostic factors identification One of the main goals of the IDMPS initiative is to evaluate patient's disease management. To do so, a key outcome in diabetes is the blood level of glycated hemoglobin (HbA1c). High HbA1c is a risk factor for micro- and macrovascular complications of diabetes [Wij+17]. Patients with T2DM who reduce their HbA1c level of 1% are 19% less likely to suffer cataracts, 16% less likely to suffer heart failure and 43% less likely to suffer amputation or death due to peripheral vascular disease [AAH19; SKP10].

Given the importance of HbA1c control for diabetic patients, we deemed interesting to focus our prognostic factors detection on patients meeting the recommended HbA1c threshold. This recommended threshold varies depending on several factors, such as age or history of vascular complications. For most T2DM patients, this threshold is set at 7%, which is how we define glycemic control for TD2M patients. Our research question can then be formulated as follows: "What are the prognostic factors of glycemic control in TD2M patients?". We consider the following variables as confounding factors: Patient's age [Ma+16], Gender [Ma+16], BMI [Can+18], Level of education [Tsh+12] and Time since diabetes diagnosis [Jua+12]. Considering the geographical heterogeneity in IDMPS, we added the continent where the data was collected.

This experiment included 1857 patients from IDMPS wave 6 and 2330 patients from IDMPS wave 7, with 63 variables considered as candidate prognostic factors. In wave 6, 17.7% of patients were under the 7% HbA1c threshold, versus 18.8% in wave 7.

Predictive factors identification Another key outcome in diabetes management is the occurrence of hypoglycemia events, which is one of the main complications linked to diabetes. Hypoglycemia symptoms include dizziness, sweating, shakiness; but can also lead to unconsciousness or death in severe cases. Previous studies have shown the impact of insulin treatments on the incidence of hypoglycemia, including comparing premixed insulin analogues to basal insulin analogues (with or without prandial insulin). In some cases, hypoglycemia rates were found to be slightly higher in patients population treated with premixed insulin analogues [Pet+18].

We focused our predictive factors detection on hypoglycemia risk in the past 3 months under premixed insulin versus basal insulin (alone or in combination with prandial insulin).

Our research question can then be formulated as follows: “What are the subgroups in which the treatment effect (premixed insulin versus basal insulin with or without prandial insulin) on the risk of hypoglycemia in the past 3 months is both positive and higher than outside the subgroups?” Illustrative example: “The risk ratio in experiencing hypoglycemia under premixed insulin versus basal insulin (with or without prandial insulin) is greater on male patients than on female patients”.

This experiment included 2006 patients from IDMPS wave 6 and 2505 patients from IDMPS wave 7, with 62 variables considered as candidate predictive factors. In wave 6, 32.4% of patients were taking Premixed insulin with a hypoglycemia rate of 32.2%, versus 25.6% for basal insulin regimen. In wave 7, 39.0% of patients were taking Premixed insulin with a hypoglycemia rate of 33.1%, versus 28.3% for basal insulin regimen.

4.2.7.2. Analytical strategies

An objective of this paper is to compare the *Q-Finder* algorithm to state-of-the-art approaches for clinical SD in both SI-SD and KDD-SD. There are a vast number of approaches in both domains, we chose two state-of-the art methods from KDD-SD to address the prognostic factors research, and two methods from SI-SD to address the predictive factors research. Among SI-SD methods, we chose SIDES (Subgroup Identification Differential Effect Search method) and Virtual Twins. The first one is arguably the most well-known local recursive methods while Virtual Twins is a recognized method, representative of global modelling approaches. In the domain of KDD-SD methods, we chose APRIORI-SD and CN2-SD which are well-known representative of respectively exhaustive and heuristic approaches to SD.

While these four methods do cover the spectrum of SD and identification methods, both SIDES and Virtual Twins are well adapted to predictive tasks, APRIORI-SD and CN2-SD can only address prognostic tasks. Since *Q-Finder* can address both tasks, it is compared with the two methods that are adapted to each of the two tasks described in section 4.2.7.1. For all the analyses, IDMPS wave 6 were used as the discovery dataset and IDMPS wave 7 as the test dataset. To allow comparison of results, only the top-10 subgroups of each algorithm are considered without any human intervention during the selection. Finally, default parameters of each

algorithm were selected, except shared parameters which we kept as similar as possible.

Exploring prognostic subgroups For each of the three approaches to identify prognostic subgroups (CN2-SD, APRIORI-SD, and *Q-Finder*) we detail the version and main parameters.

CN2-SD[14]: A beam search algorithm adapted from association rule learning CN2 to SD. It introduces a weighted covering method, where examples covered by a subgroup are not removed from the training set, but their weights are decreased. This allows examples to appear in several subgroups and cover groups with more diversity. The version used is the one found in Orange 3.23.1. The default parameters are: $WRAcc$ as the optimization metric, $beam\_width = 20$ (the bigger the beam, the more combinations are tested), $max\_rule\_length = 3$ (parameter representing the maximum complexity of a subgroup[15]) and $min\_covered\_examples = 10\%$ (minimum coverage of a subgroup[16]).

APRIORI-SD[17]: An exhaustive search algorithm adapted from association rule learning APRIORI to SD. Compared to APRIORI it only considers subgroups that contain the target variable in the right-hand side. Like CN2-SD, it also uses the weighted covering method. The Python package pysubgroup version 0.6.1 [LB18] is used, with the following parameters: $WRAcc$ as the optimisation metric, $maxdepth = 3$ [15] and $result\_set\_size\_coverage = 10\%$ [16].

***Q-Finder* prognostic mode:** The version used is 5.4 with $C_{max} = 3$, $\#Bins = 10$ and $\#Cats = \infty$ (see section 4.2.2). Only left-bounded and right-bounded intervals are considered. The thresholds for credibility criteria are the default values presented in section 4.2.3.1 : $minimum\ coverage = 10\%$, $minimum\ basic\ pattern\ absolute\ contribution = 0.2$, $maximum\ basic\ pattern\ contribution\ ratio = 5$, $minimum\ effect\ size = 1.2$ (with or without correction for confounders), and $maximum\ effect\acute{s}\ significance\ threshold = 0.05$ (with or without correction for confounders). Multiple testing correction is addressed using *Bonferroni correction* in the discovery dataset and *Benjamini-Hochberg procedure* in the test dataset. For the ranking steps, aggregation rules are the ones presented in section 4.2.3.2, m_c being the p-value for subgroup's effect when corrected for confounders. The default top-k selection is performed with the odds-ratio corrected for confounders as the second metric and $\delta_{ES} = 0.2$ (see section 4.2.4.2).

[14]https://pypi.org/project/Orange3/
[15]This corresponds to *Q-Finder*'s maximum complexity parameter
[16]This corresponds to *Q-Finder*'s minimum threshold for the coverage criterion
[17]https://github.com/flemmerich/pysubgroup

Exploring predictive subgroups For each of the three approaches to identify predictive subgroups (Virtual Twins, SIDES and *Q-Finder*) we detail the version and main parameters.

Virtual Twins[18]: Following the vignette's recommendation from the R package aVirtualTwins version 1.0.1, missing values were *a priori* imputed on the discovery dataset using $rfImpute()$ from the randomForest package version 4.6.14. For this step and each of the following, the $seed$ was set to 42. After the imputation, Virtual Twin's first step consisted in using *randomForest()* from the randomForest package (version 4.6.14) with $ntree = 500$ and $threshold = 0.5$ (threshold above which the treatment effect is considered significant for a patient). The second step consisted in performing a classification tree with $maxdepth = 3$ (maximum depth of the classification tree[15]). Only the leaves for which the predicted outcome was the target were considered as outputted subgroups.

SIDES[19]: The version considered is 1.14 from the SIDES R package. The parameters considered are: $M = 5$ (maximum number of best promising subgroups selected at each step of the algorithm), $alpha = 0.05$ (overall type 1 error rate, which is compared with p-values corrected for multiple testing using a resampling-based method to protect the overall type 1 error rate), $S = 200$ (minimum subgroup size desired, set at 10% of the discovery dataset[16]), $L = 3$ (maximum depth of the tree[15]), $D = 0$ (minimum difference between the treatment and the control arm), $gamma = 1$ (relative improvement parameter), $num\_crit = 1$ (splitting criterion used: maximizing the differential effect between the two child subgroups), $H = 1$ (i.e. no random split of the discovery dataset), $ord.bin = 10$ (number of classes continuous covariates are discretized into[20]). As SIDES is a non-deterministic algorithm, the $seed$ was set to 42.

***Q-Finder* predictive mode:** The version used is 5.4 with $C_{max} = 3$, $\#Bins$ = 10 and $\#Cats = \infty$ (see section 4.2.2). Only left-bounded and right-bounded intervals are considered. The thresholds[21] for credibility criteria are the default values presented in section 4.2.3.1 : $minimum\ coverage = 10\%$, $minimum\ basic\ pattern\ absolute\ contribution = (0, 0.2)$, $maximum\ basic\ pattern\ contribution\ ratio = (\infty, 5)$, $minimum\ effect\ size = (1, 1.2)$ (with or without correction for confounders), and $maximum\ effect\acute{s}\ significance\ threshold = (0.05, 0.05)$ (with or without correction for confounders). Multiple testing correction is addressed using *Bonferroni*

[18] https://cran.r-project.org/web/packages/aVirtualTwins/vignettes/full-example.html
[19] https://cran.r-project.org/web/packages/SIDES/index.html
[20] This corresponds to *Q-Finder*'s $\#Bins$ parameter
[21] In predictive mode the user indicates 2 thresholds instead of 1 for some criteria, with relation to the treatment effect within the subgroup (first value) and the differential treatment effect (second value)

correction in the discovery dataset and *Benjamini-Hochberg procedure* in the test dataset. For the ranking steps, aggregation rules are the ones presented in section 4.2.3.2, m_c being the p-value for differential treatment effect when corrected for confounders. Nevertheless, they are additional intermediate ranks to account for criteria with 2 thresholds (one for treatment effect within the subgroup, the other for differential treatment effect):

- **Rank i:** threshold met for treatment effect within the subgroup only
- **Rank i+1:** threshold met for differential treatment effect only
- **Rank i+2:** threshold met for both treatment effect within the subgroup and differential treatment effect

The default top-k selection is performed with the odds-ratio for differential treatment effect corrected for confounders as the second metric and $\delta_{ES} = 0.2$ (see section 4.2.4.2).

4.2.7.3. Results: Prognostic factors identification

***Q-Finder* results on the prognostic task:** *Q-Finder* generated 203 subgroups satisfying all the credibility criteria. Among the top-10 subgroups selected while accounting for diversity, 2 are of complexity 1, none are of complexity 2 and 8 are of complexity 3. The results are presented below in Table 4.1 along with the main metrics of interest computed on both the discovery and the test datasets. The two first-ranked subgroups S1 and S2 are both of complexity 1 and state that patients whose last postprandial glucose (PPG) level was below 172.0 mg/dl (resp. whose last fasting blood glucose (FBG) level was below 129.6 mg/dl) do have a better glycemic control than the others. Both subgroups are very close to the glycemic control targets established by the American Diabetes Association (resp. 180 mg/dl for PPG and 130 mg/dl for FBG [Ame17]). The coverage (or support) of the first subgroup S1 is 30% of the discovery dataset, its adjusted odds-ratio is 4.8 ($[3.5; 6.5]$) and its p-value is $1.81E-23$ on the discovery dataset. All selected subgroups were successfully reapplied on the test dataset, with odds-ratios corrected for confounders above 1.81 and p-values below 0.05 when adjusted for multiple testing by Benjamini-Hochberg procedure. It is worth noticing that all the subgroups were significant using the more conservative Bonferroni correction in the discovery dataset.

Tab. 4.1.: *Q-Finder* results on the detection of prognostic factors describing patients with better glycemic control

Subgroup Ranking*	Subgroup description	Coverage Discovery / Test	Adjusted odds-ratios (IC95%) Discovery**	p-value Discovery	Adjusted p-value Discovery***	Adjusted odds-ratios (IC95%) Test**	p-value Test	Adjusted p-value Test***
S1	Last postprandial glucose measurement (mg/dL) $\leq$ 172.0	30% / 27%	4.78 [3.5; 6.5]	1.81E-23	1.15E-18	4.28 [3.2; 5.7]	2.09E-24	1.04E-23
S2	Last fasting blood glucose measurement (mg/dL) $\leq$ 129.6	38% / 36%	3.60 [2.8; 4.7]	9.86E-21	6.28E-16	5.06 [4.0; 6.5]	9.82E-37	9.82E-36
S3	Follow healthy diet and exercise plan = Yes AND Device used for insulin: Vials and syringes = No AND Cumulated # of individual therapies taken by the patient $\leq$ 3	14% / 16%	2.57 [1.9; 3.5]	7.08E-9	4.50E-4	2.50 [1.9; 3.3]	1.78E-11	3.84E-11
S4	Follow healthy diet and exercise plan = Yes AND Device used for insulin: Vials and syringes = No AND # of different cardiovascular treatments $\leq$ 2	22% / 17%	2.26 [1.7; 3.0]	9.96E-9	6.34E-4	2.36 [1.8; 3.0]	5.24E-11	7.48E-11
S5	Follow healthy diet and exercise plan = Yes AND # of OGLD $\leq$ 1 AND Type of health insurance = Public	16% / 24%	2.47 [1.8; 3.4]	1.05E-8	6.69E-4	2.44 [1.9; 3.1]	2.20E-13	7.33E-13
S6	Follow healthy diet and exercise plan = Yes AND Covered by a health insurance = Yes AND # of different cardiovascular treatments $\leq$ 2	17% / 11%	2.34 [1.7; 3.1]	1.28E-8	8.15E-4	1.81 [1.3; 2.5]	1.42E-4	1.58E-4
S7	Follow healthy diet and exercise plan = Yes AND Covered by a health insurance = Yes AND Cumulated # of individual therapies taken by the patient $\leq$ 4	17% / 16%	2.33 [1.7; 3.1]	1.30E-8	8.27E-4	2.44 [1.9; 3.2]	1.92E-11	3.84E-11
S8	Follow healthy diet and exercise plan = Yes AND Times seen by a diabetologist in the past 3 months = 0 AND Cumulated # of individual therapies taken by the patient $\leq$ 4	16% / 17%	2.43 [1.8; 3.3]	1.85E-8	1.18E-3	2.25 [1.7; 3.0]	5.46E-9	6.82E-9
S9	Follow healthy diet and exercise plan = Yes AND Covered by a health insurance = Yes AND Treated for other form of dyslipidemia = Yes	22% / 19%	2.33 [1.7; 3.2]	1.94E-7	1.24E-2	2.64 [2.0; 3.5]	4.87E-11	7.48E-11
S10	Follow healthy diet and exercise plan = Yes AND Covered by a health insurance = Yes AND Received biguanides = No	12% / 8%	2.40 [1.7; 3.3]	2.66E-7	1.70E-2	1.86 [1.3; 2.6]	3.07E-4	3.07E-4

** Subgroup ranking is based on p-values on discovery dataset*

*** Odds-ratios are adjusted for confounding factors through multiple regression model*

**** Adjusted p-values for multiple testing are based on a Bonferroni correction (resp. Benjamini-Hochberg procedure) on the discovery dataset (resp. on the test dataset)*

Results for CN2-SD and APRIORI-SD: Results for both CN2-SD and APRIORI-SD are given below. For CN2-SD, no subgroups were outputted using the default parameters, described in 4.2.7.2. For APRIORI-SD, 186 subgroups were outputted. Among the top-10 subgroups based on the WRAcc measure, 1 is of complexity 1, 2 are of complexity 2 and 7 are of complexity 3. The complexity 1 subgroup (S4 in Table 4.2) is defined by a *last postprandial glucose measurement below 144 mg/dl* (WRAcc on discovery dataset: 0.0329). All complexity 2 and 3 subgroups, except S10, are also defined by this basic pattern, combined with other patterns such as $Receives\ GLP{-}1\ analogues = No$ or $Self{-}monitoring\ testing\ performed\ at\ bed\ time = No$. The results are presented below in Table 4.2 with the WRAcc measure, both on the discovery and the test datasets:

Tab. 4.2.: APRIORI-SD results on the detection of prognostic factors describing patients with better glycemic control

Subgroup Ranking*	Subgroup description	WRAcc Discovery	WRAcc Test
S1	Last postprandial glucose measurement (mg/dL) $\leq$ 144 AND Self-monitoring testing performed at bed time = No	3.30E-2	2.52E-2
S2	Last postprandial glucose measurement (mg/dL) $\leq$ 144 AND Self-monitoring testing performed at bed time = No AND # of sorts of required hospitalization (macro/microvascular, hypo) = 0	3.30E-2	2.47E-2
S3	Last postprandial glucose measurement (mg/dL) $\leq$ 144 AND # of sorts of required hospitalization (macro/microvascular, hypo) = 0	3.29E-2	2.82E-2
S4	Last postprandial glucose measurement (mg/dL) $\leq$ 144	3.29E-2	2.91E-2
S5	Last postprandial glucose measurement (mg/dL) $\leq$ 144 AND Self-monitoring testing performed at bed time = No AND Receives GLP-1 analogues = No	3.28E-2	2.38E-2
S6	Last postprandial glucose measurement (mg/dL) $\leq$ 144 AND Receives amylin agonist = No AND Receives GLP-1 analogues = No	3.27E-2	2.77E-2
S7	Last postprandial glucose measurement (mg/dL) $\leq$ 144 AND Receives GLP-1 analogues = No AND # of sorts of required hospitalization (macro/microvascular, hypo) = 0	3.27E-2	2.68E-2
S8	Last postprandial glucose measurement (mg/dL) $\leq$ 144 AND Receives GLP-1 analogues = Yes	3.27E-2	2.77E-2
S9	Last postprandial glucose measurement (mg/dL) $\leq$ 144 AND Self-monitoring testing performed at bed time = No AND Receives amylin agonist = No	3.27E-2	2.46E-2
S10	Follow healthy diet and exercise plan = Yes AND Receives more than 2 OGLD = No AND Patient living in = Urban area	3.08E-2	3.43E-2

** Subgroup ranking is based on WRAcc measure in discovery dataset.*

4.2.7.4. Results: Predictive factors identification

***Q-Finder* results on the predictive task:** *Q-Finder* generated 2775 subgroups in the discovery dataset that pass all the criteria of credibility on the predictive task. Among the top-10 subgroups selected while accounting for diversity, all are of complexity 3 except one. The results are presented below in Table 4.3 with main criteria of interest computed on both the discovery and the test datasets.

Subgroup S2 states that patients who use a disposable pen, don't smoke and are not heavily treated for diabetes, have a higher risk than the others in experiencing hypoglycemia under Premixed insulin than under Basal insulin (coverage = 25%, adjusted odds-ratio for differential treatment effect = 3.31 [2.0 ; 5.6], p-value = 7.13E-6).

The seven first selected subgroups were successfully reapplied on the test dataset, with adjusted odds-ratios related to differential treatment effect above 1.86. Indeed, these subgroups have a p-value below 0.05 adjusted for multiple testing using Benjamini-Hochberg procedure, even though no subgroups were "statistically

significant" after Bonferroni correction in the discovery dataset. It is worth noticing that all subgroups have adjusted odds-ratios above 1.0 in the test dataset.

Tab. 4.3.: *Q-Finder* results on the detection of predictive factors describing patients with a higher risk than the others in experiencing hypoglycemia under Premixed insulin than under Basal insulin (with or without Prandial insulin).

Subgroup Ranking*	Subgroup description	Coverage Discovery / Test	Adjusted odds-ratios for differential treatment effect (IC95%) Discovery**	p-value for differential treatment effect Discovery	Adjusted odds-ratios for differential treatment effect (IC95%) Test**	p-value for differential treatment effect Test	Adjusted p-value for differential treatment effect Test***
S1	Statins for dyslipidemia = Yes AND Device used for insulin: Vials and syringes = No AND Total # of anti-diabetics agents $\leq$ 1	28% / 31%	3.04 [1.9; 5.0]	7.02E-6	2.12 [1.4; 3.2]	2.36E-4	1.18E-3
S2	Device used for insulin: Disposable pen = Yes AND Smoking habits = Never AND Total # of anti-diabetics agents $\leq$ 1	25% / 26%	3.31 [2.0; 5.6]	7.13E-6	1.93 [1.3; 2.9]	2.04E-3	4.28E-3
S3	Total # of anti-diabetics agents $\leq$ 1 AND # of different devices used by the patient $\geq$ 1	48% / 61%	2.71 [1.8; 4.2]	9.55E-6	2.59 [1.7; 4.0]	1.92E-5	1.92E-4
S4	Treated for other form of dyslipidemia = Yes AND Times seen by a diabetologist in the past 3 months $\leq$ 1 AND Device used for insulin: Vials and syringes = No	33% / 38%	3.55 [2.0; 6.3]	1.26E-5	1.93 [1.2; 3.0]	5.02E-3	7.17E-3
S5	Receives oral glycaemic lowering drugs = Yes AND Times seen by a diabetologist in the past 3 months = 0 AND Device used for insulin: Vials and syringes = No	29% / 34%	2.98 [1.8; 4.9]	2.40E-5	1.86 [1.2; 2.8]	2.14E-3	4.28E-3
S6	Statins for dyslipidemia = Yes AND Total # of anti-diabetics agents $\leq$ 1 AND Age at diagnosis (year) $\leq$ 56	30% / 33%	2.74 [1.7; 4.4]	2.64E-5	2.04 [1.4; 3.0]	4.08E-4	1.34E-3
S7	Treated for other form of dyslipidemia = Yes AND Times seen by a diabetologist in the past 3 months $\leq$ 1 AND # of different devices used by the patient $\geq$ 1	33% / 44%	3.37 [1.9; 6.0]	2.79E-5	2.05 [1.2; 3.4]	4.58E-3	7.17E-3
S8	Statins for dyslepidemia = Yes AND Device used for insulin: Vials and syringes = No AND HDL serum cholesterol (mg/dL) $\leq$ 58.0	27% / 30%	3.22 [1.9; 5.6]	2.82E-5	1.05 [0.7; 1.7]	8.21E-1	8.21E-1
S9	Statins for dyslipidemia = Yes AND Visits diabetes websites = No AND Duration of insulin therapy (year) $\geq$ 4	34% / 32%	2.59 [1.7; 4.1]	3.12E-5	1.14 [0.8; 1.7]	5.09E-1	5.65E-1
S10	Other form of dyslipidemia = Yes AND Visits diabetes websites = No AND Duration of insulin therapy (year) $\geq$ 4	40% / 37%	2.56 [1.6; 4.0]	3.22E-5	1.25 [0.9; 1.8]	2.48E-1	3.10E-1

** Subgroup ranking is based on p-value for differential treatment effect on discovery dataset*

*** Odds-ratios are adjusted for confounding factors through multiple regression model*

**** Adjusted p-values for multiple testing are based on a Benjamini-Hochberg procedure on the test dataset*

Results for SIDES and Virtual Twins on the predictive task: Results for both SIDES and Virtual Twins are given below. For SIDES, no subgroups were outputted using the default parameters, described in section 4.2.7.2. For Virtual Twins, only three subgroups were obtained, 1 of complexity 2 and 2 of complexity 3. The results are presented below in Table 4.4 with the metrics that are outputted from the algorithm, both on the discovery and the test datasets. All subgroups are defined by a same attribute, the "number of different lipid-lowering agents for dyslipidemia".

Tab. 4.4.: Virtual Twins results on the detection of predictive factors describing patients with a higher risk than the others in experiencing hypoglycemia under Premixed insulin than under Basal insulin (with or without Prandial insulin).

Subgroup Ranking*	Subgroup description	Treatment event rate Discovery / Test	Control event rate Discovery / Test	Treatment sample size Discovery / Test	Control sample size Discovery / Test	Risk Ratio Discovery / Test
S1	# of OGLD ≥ 2 AND # of different lipid-lowering agents for dyslipidemia ≥ 1	33% / 34%	16% / 26%	72 / 408	340 / 755	2.06 / 1.36
S2	# of OGLD ≤ 2 AND Duration of insulin therapy (year) ≥ 3 AND # of different lipid-lowering agents for dyslipidemia ≥ 1	38% / 39%	29% / 32%	238 / 339	432 / 433	1.31 / 1.21
S3	Receives oral glycaemic lowering drugs = Yes AND Total serum triglycerides (mg/dL) ≥ 169.7 AND # of different lipid-lowering agents for dyslipidemia $= 0$	24% / 33%	23% / 27%	57 / 9	110 / 18	1.08 / 1.20

** Subgroup ranking is based on risk ratios in discovery dataset*
In the discovery dataset, risk ratios were computed after missing values imputation. In the test dataset, risk ratios were computed on the original dataset.

4.3 Applications to metagenomics for phenotype status prediction

The subgroup analysis described above with the *Q-Finder* has demonstrated its advantages in terms of interpretability and statistical robustness on an observational study problem. SD does not try to cover the entire database regarding prediction, which prevents this approach from being used as a classification algorithm. Therefore, personalized prediction can only be done on patients who are in subgroups. It is necessary to reformulate the algorithm so that it can be used to classify metagenomic data based on taxa abundance. We want to combine the individual stratification aspect to the explicability of the model predictions. This section details the *Q-Classifier*, a SD algorithm adapted to classification task. First, the concepts of the algorithm are introduced, then the redesigned statistical metrics and optimization are described, next the rejection and delegation operations are discussed, and finally, the results obtained on several datasets are presented.

4.3.1 Overview and concepts of the *Q-Classifier*

The *Q-Classifier* is a multi-class supervised learning algorithm that generates statistically credible subgroups to discriminate examples in different classes. It combines subgroup analysis with supervised learning because some examples have an interpretable prediction when they can be predicted by rules, while the more complex ones are predicted by a state-of-the-art supervised algorithm (SVM, Random Forest, etc ...). In this sense, it can be seen as a reject and cascade classifier.

The algorithm is in line with precision medicine because it may create different interpretations adapted for each individuals as they may respond differently to a given pathology. Moreover, no example is excluded from the prediction as it is the case for other SD algorithms. This makes it possible to have total coverage and not to reduce the classification scores. Therefore, it is necessary that generation and aggregation of rules must be computed on all classes, representing controls and cases in our metagenomic datasets. The rules generation is done with the *Q-Finder*, while the rules refinement keeps the same scheme (credibility metrics and subgroup ranking) but the steps have been modified for the classification task and are described in the next section 4.3.2. It should be noted that, in section 4.1.2, two types of SD analysis have been defined, namely SI and KDD, handled by the algorithm *Q-Finder*. Since we are in a case where there is no treatment arm on the analyzed metagenomic data, only the KDD part of the *Q-Finder* is useful for building the model.

Two data preprocessing options could be included in the algorithm. First, a dimensionality reduction operation to limit the number of rules generated by the *Q-Finder*, which reduces runtime and improves statistical power when adjusting the p-value with the Bonferroni correction. Indeed, one of the weaknesses of the *Q-Finder*, and thus of the *Q-Classifier*, is its high complexity which is equal to $O(G \times (F \times (M \times D)^C)$. Where C is the rule complexity, F is the aggregation rules complexity, G is the number of groups (e.g., control / case), D is the number of variables and M is the maximum of modalities per variable. In practice, the recursive feature elimination with a SVM model is fitted to perform the features reduction. The second processing step is related to the nature of the compositional data and can be done by a log-ratio transformation. As explained in section 1.1.6, the only log-ratio transformation that does not alter the dimensionality, which is necessary to retain the variable names and produce an interpretable prediction, is the centered log-ratio transformation (CLR). The *Q-Classifier* algorithm is summarized in Figure 4.5 from metagenomic sequencing to the training and the final prediction.

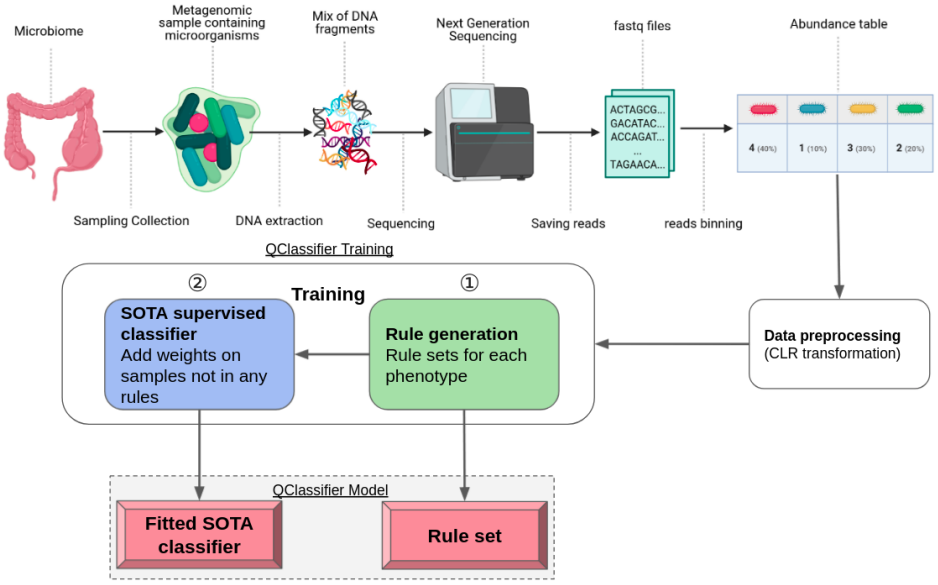


Fig. 4.5.: *Q-Classifier* overview: The algorithm takes as input the calculated metagenomic abundance data and starts by preprocessing according to the selected parameters (such as CLR transformation). The training phase is composed by one step of statistically credible subgroups generation followed by state-of-the-art classifier training. At the end, the algorithm consists of a set of rules and a state-of-the-art classifier cascaded during the classification step.

4.3.2 Statistical metrics and optimal union

Statistical metrics The main modification of the metrics is to evaluate the subgroup classification scores rather than the confidence score (risk ratio). The list below enumerates the new credibility criteria and thresholds used to filter subgroups in the *Q-Classifier* model:

1. **Coverage criterion:** Same as in section 4.2.3.1, but the default minimum threshold is now 20% to prevent overfitting as the metagenomic datasets are after small (few hundreds of samples).

2. **F1-score criterion:** A more relevant classification score than accuracy on unbalanced data that combines precision and recall metrics so that they have equal relative contributions. The formula is defined by $\frac{2\times TP}{2\times TP+FN+FP}$ with TP as true positive, FN as false negative and FP as false positive. The default lower threshold for this criterion is set to 0.5.

3. **Basic pattern contribution criterion:** Similar to the one described in section 4.2.3.1 but defined to control the F1-score contribution of the basic patterns.

Default thresholds are arbitrarily set to be greater than 0.03 for the *absolute contribution* and lower than 1.05 for the *contribution ratio*.

4. **Effect significance criterion:** This criterion is replaced by the hypergeometric p-value which is computed as the sum of the mass probability functions of the hypergeometric law in the interval $[TP, TP + FP]$. This discrete law is chosen because it is well adapted to the analysis of the confusion matrix from a prediction model. The default threshold value is still defined to at most 5%
5. **Effect adjusted significance criterion:** The Bonferroni correction is applied to control the type 1 errors. The hypergeometric p-value is multiplicated by the number of generated subgroups and the threshold remains 5%.

The aggregation rules is similar to the one in section 4.2.3.2 but in the *Q-Classifier* the criteria defined above are used:

- Rank 1: subgroups that satisfy the coverage criterion
- Rank 2: subgroups of rank 1 that also satisfy the F1-score criterion
- Rank 3: subgroups of rank 2 that also satisfy the basic patterns contribution criterion
- Rank 4: subgroups of rank 3 that also satisfy the significance criterion
- Rank 5: subgroups of rank 4 that also satisfy the effect adjusted significance criterion

Optimal union This approach, whose pseudo-code is written in the algorithm 4, consists in creating a set of the best subgroups, from the set of all subgroups having passed the rank 5, which maximizes a predefined metric (in our case the F1 score). The optimal union differs from the top-k selection algorithm (section 4.2.4.2) used in *Q-Finder* algorithm which increases the diversity of the basic patterns.

The *optimal_union* (algorithm 4) is initialized with the subgroup that has the best score according to the chosen metric. It then performs two steps called “forward” and “backward”. The “forward” phase searches for the rules which, once added, improve the score of the union the most. If no improvement can be made by the “forward” phase, then the algorithm ends. Otherwise, the “backward” phase is executed after the addition of the new rule to check if, on the contrary, removing a rule (except the one with the best score) would improve the score of the union. Any rule that has been added to the optimal union is no longer considered afterwards

Algorithm 4: *optimal_union*: From the subgroups that pass the rank 5 of the credibility criteria, takes those that maximize a specific metric (*metric_to_maximize*)

input : $\boldsymbol{G_{metric}}$: Set of subgroups with computed metrics
metric_to_maximize: Metric maximized in optimal union
nb_rules_max: (Default=10) Number maximum of rules in the optimal union
threshold: (Default=0.001): Minimum gain when adding a rule in the optimal union
output : $\boldsymbol{G_{opti}}$: Set, optimal union of subgroups in $\boldsymbol{G_{metric}}$

G_{opti} = {}
score_prev_forward = -inf
forward_done = False
while *not forward_done* **do**
    // Case where G_{opti} is higher than k we stop
    if $G_{opti} \geq$ *nb_rules_max* **then**
        break
    end
    /* Forward: Compute scores for each rule in the set if they are added in the optimal union (G_{opti}) */
    scores = [ ]
    for *rule in* G_{metric} **do**
        scores.append(metric_to_maximize(G_{opti} U {rule}))
    end
    // best_score return the best score in scores
    score_forward = best_score(scores)
    if *score_forward - score_prev_forward* $<$ *threshold* **then**
        forward_done = True
    else
        // best_rule return the rule with the best score in scores
        G_{opti}.append(best_rule(scores))
        score_prev_forward = score_forward
        /* Backward: Compute scores for each rule in the optimal union (G_{opti}) if they are removed from it */
        backward_done = False
        while *not backward_done* **do**
            backward_done = True
            // There is at least 3 rules
            if *length*(G_{opti}) $>$ *2* **then**
                /* Do not look at the first rule (index 0) to ensure not removing the best rule */
                scores = [ ]
                for *rule in* G_{opti}*[1:]* **do**
                    scores.append(metric_to_maximize($G_{opti} \setminus$ {rule}))
                end
                score_backward = best_score(scores)
                if *score_backward* $>$ *score_forward* **then**
                    G_{opti}.remove(best_rule(scores))
                    backward_done = False
                end
            end
        end
    end
end

even if it has been removed by a backward phase. We iterate the process from the forward phase until the algorithm ends (i.e., no rule improves the optimal union).

4.3.3 Rejection and delegation concepts to adapt SD for prediction

To combine SD and supervised learning, the *Q-Classifier* does not make a majority vote between models or an average of the probability prediction, but it forms a cascading prediction that follows these principles:

a. All samples present in at least one subgroup or in several subgroups with the same class are stratified by the SD approach with an interpretable prediction.

b. All samples not assigned to any subgroups or in several subgroups with different classes are rejected and delegated to a classifier.

In the training step, the state-of-the-art classifier uses the entire training set. However, the algorithm adds a higher weight to the samples rejected in the SD stage (case **b**) in order to reinforce its learning on the most difficult examples to classify. The weights are arbitrary set to 3 times higher compared to accepted samples (case **a**). In the classification step, samples in case **a** are only predicted by the SD approach and are excluded from the classifier prediction, while samples in case **b** are first attempted to be predicted by the SD approach and then rejected for delegation to the state-of-the-art classifier. The training and classification steps are summarized in figures 4.6 and 4.7 respectively, and in algorithms 5 and 6 respectively.

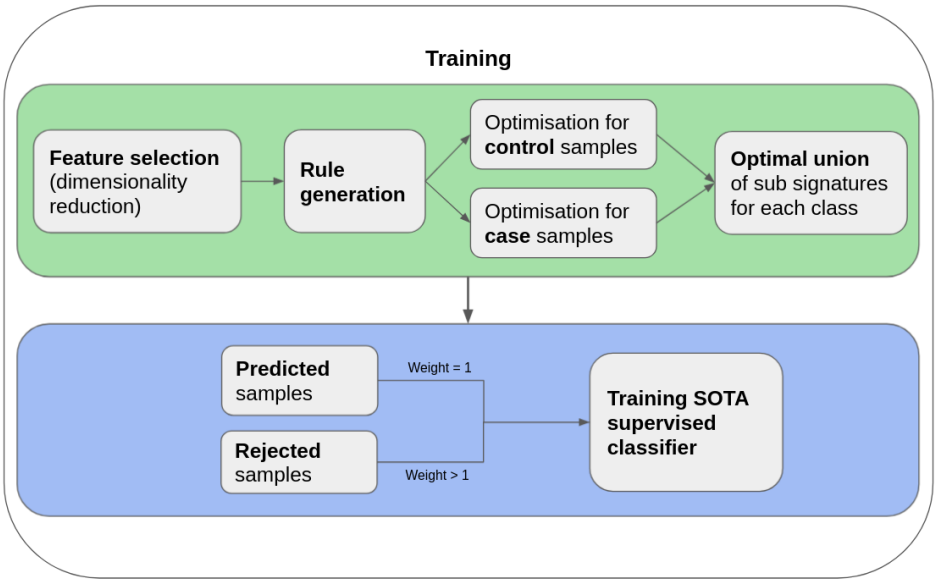


Fig. 4.6.: *Q-Classifier* training stage: An optional feature selection is first processed, then statistically credible subgroups on all classes (control and case) are generated. Optimal unions of metagenomic sub-signatures for each class are computed and gathered. Finally, a SOTA classifier is trained by adding more weight to the data that has been rejected.

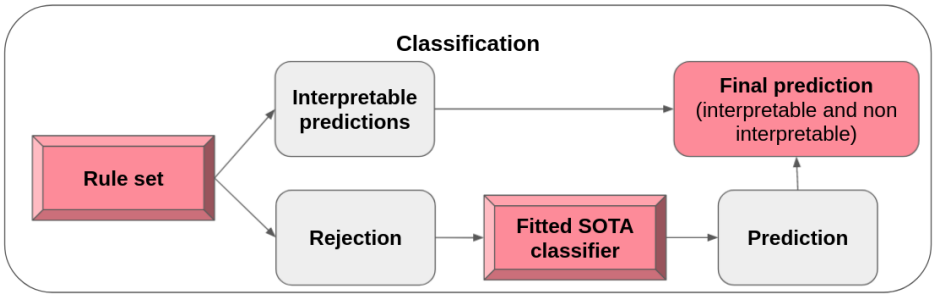


Fig. 4.7.: *Q-Classifier* classification stage: samples which are not rejected by the rule set have therefore an interpretable prediction while the rejected ones are predicted by a fitted SOTA classifier.

4.3.4 Benchmark on real-world and simulated metagenomic data

The four real-world and the two simulated datasets used during the experiments of the *Q-Classifier* are detailed in sections 2.1 and 2.2 respectively. Each result is calculated by 10-fold cross-validation with 80% in train and 20% in test. The length of the rules (complexity) of the *Q-Classifier* is set to 2 and the default parameters are

Algorithm 5: *Q-Classifier* training stage

input : **X**: Matrix, train data
y: Vector, train label
algo_subgroup: Algo, to generate subgroups
algo_metrics: Algo, to compute some metrics over subgroups
SOTA: Algo, a SOTA classifier
metric_to_maximize: Metric maximized in optimal union
feature_reduction: Algo, to keep only best features
transformation: Algo, to process raw data (e.g: CLR)
weight_reject: Float, weight on rejected to train SOTA

output : G_{opti}: Dictionary, optimal union of subgroups for each class
SOTA: Algo, a fitted SOTA classifier

if *feature_reduction* **then**
  X = recursive_feature_elimination(X)
end
if *transformation* **then**
  X = tranformation(X, min_feature)
end
$G = algo\_subgroup(X)$ `// Set of generated subgroups`
$G_{opti} = dict()$
`// Compute subgroup metrics for each class`
for *class in unique(y)* **do**
  G_{metric} = algo_metrics(X, class, G)
  $G_{opti}[class]$ = optimal_union(G_{metric}, metric_to_maximize)
end
`// predict: function that assigns subgroups to samples`
$X_{subgroup\_pred}$, X_{reject} = predict(X, G_{opti})
`// Add more weights on rejected samples`
weights = []
for *x in X* **do**
  if *x in* $X_{subgroup\_pred}$ **then**
    weights.append(1)
  else
    weights.append(weight_reject)
  end
end
SOTA.fit(X, y, weigths)

Algorithm 6: *Q-Classifier* classification stage

input : **X**: Matrix, test data processed in the same way as train data
$\boldsymbol{G_{opti}}$: Dictionary, optimal union of subgroups for each class
SOTA: Algo, a fitted SOTA classifier

output : $\boldsymbol{X_{pred}}$: Matrix, the X matrix with a prediction for each sample

// $X_{subgroup\_pred}$ contains samples with interpretable prediction

// X_{reject} contains samples that have been rejected

1 $X_{subgroup\_pred}$, X_{reject} = predict(X, G_{opti})

// Use only the rejected samples for the prediction

2 X_{sota_pred} = SOTA.predict(X_{reject})

// The final prediction is the concatenation of both prediction from rules and SOTA

3 X_{pred} = concatenate($X_{subgroup\_pred}$, X_{sota_pred})

used for its rule refinement part (see section 4.3.2), while a random search is performed to tune the parameters of the SOTA classifiers (SVM, Random Forst, Gradient Boosting and Ada Boost) and the one with the best scores is selected. Feature selection is performed by recursive feature elimination trained by 3-fold cross-validation with a minimum of 40 features kept. *MetaML* [Pas+16] and *Predomics* [Pri+20] are the reference methods. Four of our approaches are experimented on all datasets, the *Q-Classifier* is trained either on the *MetaPhlAn2* species abundance data from the Pasolli et al. [Pas+16] study or on those from the *FastDNA* [MV19], transformed or not by CLR. The reference methods are the same as in the section 3.4.2. Results on real-world and simulated datasets are summarized in Table 4.5 and 4.6 respectively. The initial rejected rate (IRR) term defines, in the tables, the percentage of rejected and delegated samples (case **b** in section 4.3.3) by the *Q-Classifier*. Accuracy, F1-score, Precision and Recall are the computed metrics. The standard deviation is calculated for the accuracy score and written with the symbol $\pm$.

4.3.4.1. Analysis of the results of the real-world datasets

Method	Metrics	*Colorectal*	*Cirrhosis*	*Obesity*	*T2D*
MetaML	Accuracy	0.81 ±0.068	0.88 ±0.043	0.64 ±0.028	0.66 ±0.052
	Precision	0.82	0.89	0.54	0.67
	Recall	0.81	0.88	0.64	0.66
	F1-score	0.79	0.88	0.54	0.66
Predomics	Accuracy	-	0.84 ±0.035	0.66 ±0.035	0.68 ±0.030
Q-Classifier MetaPhlAn2	Accuracy	0.57 ±0.074	0.88 ±0.046	0.62 ±0.010	0.65 ±0.062
	Precision	0.61	0.95	0.66	0.64
	Recall	0.8	0.81	0.84	0.67
	F1-score	0.69	0.87	0.74	0.65
	IRR	0.62	**0.17**	0.87	0.76
Q-Classifier MetaPhlAn2 CLR	Accuracy	**0.85** ±0.047	**0.94** ±0.049	**0.79** ±0.033	**0.81** ±0.053
	Precision	**0.84**	**0.97**	**0.81**	**0.80**
	Recall	**0.93**	**0.91**	**0.89**	0.80
	F1-score	**0.88**	**0.94**	**0.85**	**0.80**
	IRR	**0.54**	0.20	**0.76**	0.77
Q-Classifier FastDNA	Accuracy	0.70 ±0.049	0.81 ±0.036	0.64 ±0.048	0.68 ±0.084
	Precision	0.61	0.79	0.67	0.64
	Recall	0.56	0.88	0.87	0.74
	F1-score	0.58	0.83	0.76	0.68
	IRR	0.63	0.73	0.8	0.70
Q-Classifier FastDNA CLR	Accuracy	0.75 ±0.075	0.86 ±0.023	0.71 ±0.076	**0.81** ±0.023
	Precision	0.66	0.88	0.76	0.79
	Recall	0.67	0.88	0.84	**0.82**
	F1-score	0.66	0.88	0.79	**0.80**
	IRR	0.61	0.28	0.8	**0.54**

Tab. 4.5.: Classification results on four real-world benchmark datasets (Table 2.2). Results are reported for two reference methods (*MetaML* and *Predomics*). *Q-Classifier* with *MetaPhlAn2* or *FastDNA* abundance and with CLR transformation of not are our methods tested in this experiments.

Classification performances for disease prediction from metagenomic sample abundance: The best approach is the *Q-Classifier* on *MetaPhlAn2* data with CLR transformation reaching higher scores for almost all metrics in real-world datasets. The

CLR transformation systematically improves the scores either from *MetaPhlAn2* or from *FastDNA* abundances. We can notice that the deep learning approach *FastDNA* combined with the *Q-Classifier* gets comparable results to the *MetaPhlAn2* structuring without CLR transformation, and slightly lower scores using CLR transformation excepted on the *T2D* dataset where they are equivalent. The IRR value varies from one method and dataset to another ranging from 17% to 87%. We note that the IRR remains quite high and thus the samples predicted by a SOTA classifier represent more than half of the colorectal dataset and more than three quarters of the obesity and T2D datasets. This represents a weakness of the algorithm because a majority of samples do not have interpretable predictions. To strengthen the understanding of the predictions made by the SOTA classifiers, one method (not tested in our experiments) could be to use an interpretability tool such as the Shapley values [LL17] which give the contribution of features to the prediction of a sample relative to the average prediction of the dataset.

Metagenomics signatures as rules On the *Cirrhosis* dataset using a CLR transformation, *Q-Classifier* generated 1287 rules on the training set (80% of the 232 subjects) which passed the 5 criteria, 1130 are of complexity 1 and 157 are of complexity 2. The optimal union reduces this set of rules by taking only 5 rules of complexity 1. These rules are listed below (RCO (resp. RCA) refers to the rules of the control (resp. case) samples) with their metrics computed on the validation set (47 subjects: 23 controls and 24 cases) and adjusted for multiplicity (FDR) :

RCA1 If *Veillonella Unclassified* ≥ 4.42, then class is **Case** with $coverage = 53\%$, $F1\text{-}score = 0.94$ and $p\text{-}value = 1.23 \times 10^{-9}$

RCA2 If *Streptococcus Parasanguinis* ≥ 4.92, then class is **Case** with $coverage = 26\%$, $F1\text{-}score = 0.67$ and $p\text{-}value = 5.18 \times 10^{-5}$

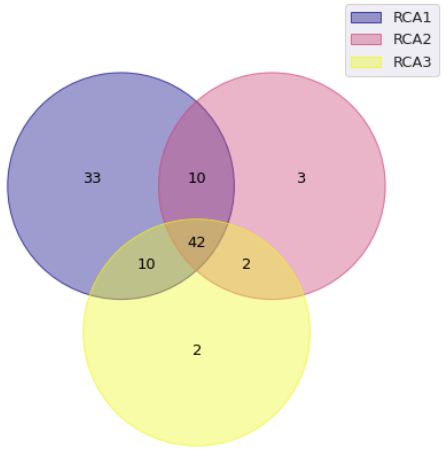
RCA3 If *Streptococcus Anginosus* ≥ 3.13, then class is **Case** with $coverage = 26\%$, $F1\text{-}score = 0.67$ and $p\text{-}value = 5.18 \times 10^{-5}$

RCO1 If *Veillonella Unclassified* ≤ 5.40, then class is **Control** with $coverage = 55\%$, $F1\text{-}score = 0.90$ and $p\text{-}value = 3.93 \times 10^{-8}$

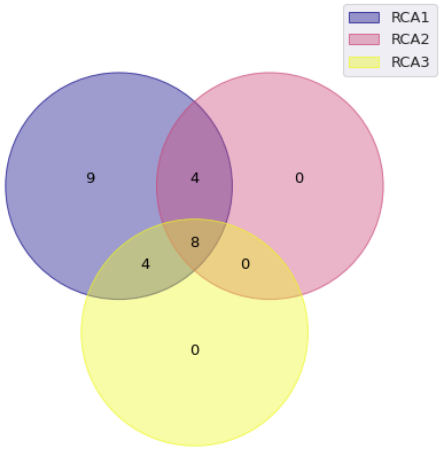
RCO2 If *Veillonella Dispar* ≤ 1.67, then class is **Control** with $coverage = 38\%$, $F1\text{-}score = 0.78$ and $p\text{-}value = 1.53 \times 10^{-5}$

Each of these rules represent a metagenomic signature of either case or control samples. When a patient is not covered by any rule it is delegated to a default classifier as explained above. Compared to the interpretable model provided by

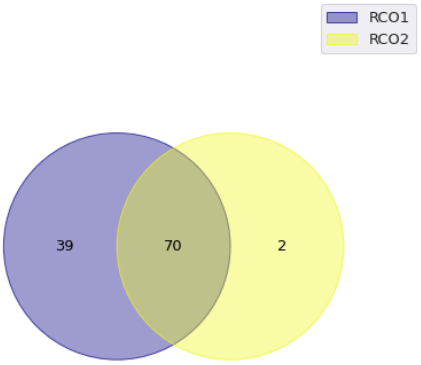
predomics from the study of Prifti et al. [Pri+20], there are 3 species in common: *Veillonella Unclassified*, *Streptococcus Anginosus* and *Veillonella Dispar*. Although *predomics* models are formula (sums, difference or ratio) of abundance, the fact that both *predomics* and *Q-Classifier* share critical bacterial species proves a level of consistency as some of these bacterial species are known to be related to the disease (here cirrhosis). Figure 4.8 provides a Venn diagram visualization of the disjunction and union of the subgroups described above.



(a) Case samples on training set.



(b) Case samples on validation set.



(c) Control samples on training set.

(d) Control samples on validation set.

Fig. 4.8.: Venn diagram of the subgroups in the optimal union of the *Cirrhosis* dataset. Each circle corresponds to a subgroup characterized by a rule. The values inside the circles correspond to the number of samples in the subgroups. When a value lies between several circles, it represents the number of samples shared by the corresponding subgroups. 161 (resp. 41) of the 186 (resp. 46) samples in the training set (resp. validation set) are covered by the union, 74 (resp. 5) of them are rejected and delegated.

The best rule for the control class and the case class are respectively “RCO1” and “RCA1”. We notice that the subgroups associated with these rules have a high intersection with the other subgroups in the optimal union. This is even more important on the validation set where sometimes one subgroup is completely included in another. As the optimal union of the rules is computed on the training set, it is possible

in validation to obtain these results. This visualization allows us to determine the disjunction or union of the subgroups' samples.

4.3.4.2. Analysis of the results of the simulated datasets

Method	Metrics	*Null Model Nanopore*	*Ecological Model Nanopore*
Q-Classifier Initial abundance	Accuracy	**0.86** ±0.017	0.82 ±0.022
	precision	**0.89**	0.84
	recall	0.83	0.83
	F1-score	**0.86**	0.81
	IRR	0.38	0.57
Q-Classifier Initial abundance CLR	Accuracy	0.83 ±0.037	**0.87** ±0.012
	precision	0.84	**0.85**
	recall	0.83	**0.90**
	F1-score	0.83	**0.87**
	IRR	0.39	**0.51**
Q-Classifier FastDNA	Accuracy	0.84 ±0.017	0.68 ±0.020
	precision	0.85	0.65
	recall	**0.86**	0.75
	F1-score	0.85	0.69
	IRR	0.46	0.73
Q-Classifier FastDNA CLR	Accuracy	0.83 ±0.038	0.66 ±0.036
	precision	0.85	0.64
	recall	0.83	0.75
	F1-score	0.84	0.69
	IRR	**0.37**	0.75

Tab. 4.6.: Classification results on two simulated datasets (Table 2.3). The initial abundances (the first two models) correspond to the simulated abundance tables before being run through a simulator. The abundances computed by *FastDNA* (The last two methods) are obtained after simulating the reads with the *Nanosim* software [Yan+17] and predicting their class to recover the initial abundance.

Classification performance for classifying simulated metagenomic samples On the *Null Model Nanopore* dataset (section 2.2.2.1), the scores are very close for each approach but on *Ecological Model Nanopore* dataset (section 2.2.2.2) the *Q-Classifier* with *FastDNA* does not reach the performance compared to the initial simulated abundance tables. This is because this dataset was created by altering the abundance

of some species guilds and during the simulation and binning this information was not fully recovered which undeniably impacted the results.

The altered species abundances and the multiplicative factors are described below:

- Case samples:
  - Profile 1: Streptococus vestibularis (0.51) and Akkermansia muciniphila (31.06)
  - Profile 2: Granulicatella adicens (38.27) and Bifidobacterium dentium (0.51)
- Control samples:
  - Profile 1: Prevotella timonensis (77.74) and Fusobacterium nucleatum (0.50)
  - Profile 2: Solobacterium moorei (7.15) and Turicibacter sanguinis (0.54)

We summarize in Table 4.7 the number of different taxa retrieved by the model's rules at the species and genus level. We can observe that all rules of the *Q-Classifier* on initial abundance with CLR retrieve a species that has been altered by one of the specific profiles. It also generates a rule of complexity 2 that reconstruct a complete profile. Regarding other methods, few species and genomes are retrieved and only rules of complexity passed all the credibility criteria of the algorithm. We deduce that for the ecological simulation (e.g., with a holistic vision), the CLR transformation seems to be very well adapted to create relevant rules. However, after the simulation of the reads, the combination of *Q-Classifier* and *FastDNA* still has difficulties in finding the patient profiles in this specific simulated dataset, which may explain the difference in results between the approaches. Rules generated by the *Q-Classifier* on initial abundance with CLR are written bellow:

RCA1 If Prevotella Timonensis ≤ 1.78, then class is **Case** with $coverage = 66\%$, $F1\text{-}score = 0.66$ and $p\text{-}value = 2.96 \times 10^{-3}$

RCA2 If Fusobacterium Nucleatum ≥ 2.28, then class is **Case** with $coverage = 53\%$, $F1\text{-}score = 0.69$ and $p\text{-}value = 2.55 \times 10^{-7}$

RCO1 If Fusobacterium Nucleatum ≤ 2.28 and Prevotella Timonensis ≥ -1.41, then class is **Control** with $coverage = 37\%$, $F1\text{-}score = 0.74$ and $p\text{-}value = 1.36 \times 10^{-16}$

RCO2 If Solobacterium Moorei ≥ -1.32, then class is **Control** with $coverage = 41\%$, $F1\text{-}score = 0.52$ and $p\text{-}value = 5.68 \times 10^{-2}$

we can notice that the rule with the best F1-score and p-value is the one of complexity 2 which is relevant to the way the simulated profiles are created. The appendix provides all the box plots showing the differences between the classification scores obtained by the SOTA classifiers alone, the *Q-Classifier* without SOTA, and the cascaded combination of the two algorithms.

Method	# Species	# Genus	# Species by rule	# Genus by rule	class	Rule's complexity
Q-Classifier Initial abundance	0	1	0	1	Control	2
			0	1	Control	1
			0	1	Case	1
			0	0	Case	0
Q-Classifier Initial abundance CLR	5	5	1	1	Control	1
			1	1	Control	1
			2	2	Case	2
			1	1	Case	1
Q-Classifier FastDNA	1	1	0	0	Control	1
			0	1	Control	1
			0	1	Control	1
			1	1	Case	1
			0	1	Case	1
			0	1	Case	1
Q-Classifier FastDNA CLR	0	2	0	1	Control	1
			0	0	Control	1
			0	0	Control	1
			0	1	Case	1
			0	1	Case	1
			0	1	Case	1
			0	1	Case	1

Tab. 4.7.: Number of taxa retrieved by species and genus for each model

4.4 Conclusion

In this chapter, we studied a category of interpretable models in the perspective of using them in precision medicine in the field of metagenomics. *Q-Finder* algorithm is a subgroup discovery method belonging to the KDD and SI family. We have improved its rule generation step that allows the algorithm to create statistically credible subgroups. It has been benchmarked against state-of-the-art algorithms like *APRIORI-SD*, *CN2-SD*, *Virtual Twins* or *SIDES* and has had better performances on rule metrics and statistical significance. We have ended its use in the context of metagenomic data. The *Q-Finder* was transformed and adapted to a classifier model called *Q-Classifier* to create personalized and interpretable stratification on metagenomic data. The *Q-Classifier* algorithm is formed as a cascading classifier of state-of-the-art subgroup discovery and classifier models. We are also interested in considering the compositional characteristic of the data, which may be neglected in

some studies, by applying a CLR transformation that allows to preserve the structure of the data necessary for the interpretability. This preprocessing step effectively improved the results obtained in the experiments. On the 4 real-world datasets, the *Q-Classifier* reached comparable or superior performances to the state-of-the-art while bringing more interpretability for samples classified by rules. Moreover, two simulated datasets have been used to confirm the ability of the algorithm to build relevant subgroups on *Null Model* and *Ecological* metagenomic simulation approach. Nevertheless, the *Q-Classifier* approach is not fully interpretable as in some cases it delegates the classification to a SOTA classifier. As such it represents a trade off between accuracy and interpretability in learning from metagenomic data. One perspective is to test interpretable tools, such as the Shapley values [LL17], to analyze the contribution of the features for the prediction of the SOTA classifiers. Finally, during our experiments, the *Q-Classifier* algorithm was tested by providing as input the abundances generated by the deep learning algorithm *FastDNA* and obtained promising results although slightly worse than those obtained by the software *MetaPhlAn2*. This means that the *Q-Classifier* method could be used in an end-to-end way without having to use gene catalogs.

5 Conclusion and perspectives

5.1 Summary of contributions

Advances in sequencing technologies over the past two decades, allowing for high-throughput, scalable and rapid sequencing, have made it possible to generate large quantities of omics data. This has accelerated the developpment of many Disciplines, including genomics, transcriptomics, proteomics and metagenomics. This thesis focused on metagenomics, a field to study the composition and interactions of taxa present in a given environment. Many studies have developed metagenomic bioinformatic techniques to process the large volume of data to better determine the presence of different taxa, their relationships in the same ecosystem and their impact on human phenotype/disease. Machine learning algorithms have produced valuable results in such applications, but several problems remain to be solved. Indeed, existing methods rely on complex workflows composed of distinct steps (fastq file cleaning, sequence assembly, sequence alignment and sequence classification) that rely on assumptions impacting the final outcome. This makes these workflows difficult to reuse from one study to another. Some do not meet the criteria of "point-of-care" processing due to the computational time and resources required to infer the results. Others are based on black box models that do not offer sufficient interpretability, restricting their use in the context of precision medicine that favors explicability. In this work we made two main contributions that address these two issues.

End-to-end deep learning from raw metagenomic data We focus on the use of deep learning algorithms to develop our approach named *Metagenome2Vec*. It aims at creating a more suitable condensed representation, called embedding, of metagenomes from their raw DNA sequences in order to be used to classify the phenotype (in our case a disease) of the samples. Once the weights of the neural networks are learned, the model has “stored” the information necessary to perform a downstream task (prediction, clustering, ...). The use of external resources, such as gene or genome catalogs, is no longer necessary for inference as it is the case with standard metagenomics workflows [PGB20]. Our approach may therefore emerge as a solution to

point-of-care processing in metagenomics, as results are obtained in about one hour with 24-48 CPUs for one sample.

We determine different experiments to carry out the validation of *Metagenome2Vec*. The significance of *Metagenome2Vec* is evaluated with intrinsic and extrinsic evaluations to control the learning of the model. When bad results are obtained during the first tests, then the training of the algorithm can be stopped to save time. Indeed, the metagenomic data being voluminous (several terabytes), the DL model having millions of weights to train and several hyperparameters to optimize, the learning time is computationally intensive (variable between 1 and 5 days depending on the computing resources and the defined parameters). Next, we benchmark *Metagenome2Vec* versus the state-of-the-art algorithms that use bioinformatics pipelines to build abundance tables [Pas+16; OZ20]. Our best model is defined by training supervised state-of-the-art algorithms (such as *SVM*, *Random Forest*, *Gradient Boosting*, ...) on the concatenation of metagenome representations learned by *M2V-Abundance* and *M2V-MIL-VAE*. The first representation corresponds to the species abundances computed by the *Read2Genome* part of the pipeline with the *FastDNA* algorithm and the second is an embedding learned by a Variational Auto-Encoder on multiple instances of metagenome embeddings. The results demonstrate the interest of the creation of DNA embeddings as well as the multiple instance learning which allowed to obtain performances equivalent or superior to the state-of-the-art approaches on the 4 Illumina real-world data sets. Finally, we verify that our approach performs well on simulated data from the 3rd generation Nanopore sequencer and thus that the model is able to learn from long reads.

Subgroup discovery for credible metagenomic signature Subgroup discovery algorithms provide interpretable results naturally using simple formulas or equations, without the need for post-hoc interpretation as in black box models. This is an interesting solution for precision medicine, as it delivers actionable information and can be useful in metagenomics, such as for fecal transplantation. A first work consists in exploring the *Q-Finder* algorithm developed by Quinten in order to enhance the statistical credibility of its subgroups and benchmark it in the subgroup analysis literature. Then, we modify its functioning to create an algorithm, named *Q-Classifier*, for metagenome classification. It is a hybrid model returning, for a set of samples, an interpretable prediction with subgroup discovery approach and classifying the remaining samples with a supervised algorithm. Trained on taxon abundance tables, *Q-Classifier* is able to generate rules explaining which taxa may possibly play a role in the prediction. We also consider the compositional nature of the abundance tables, applying specific log-ratio transformation such as CLR. Experiments highlight the

predictive and interpretability power of the model, especially when using the CLR transformation. The *Q-Classifier* obtains its best performances in disease prediction from metagenomic data when it is trained on species abundance table computed by *MetaPhlAn2* outperforming the state-of-the-art methods. The algorithm is also able to retrieve the species involved in the phenotypic class defined in the simulated dataset, showing its strength in learning credible subgroups. Moreover, *Q-Classifier* is tested with the species abundance tables from our previous end-to-end deep learning approach and the results, although slightly inferior, remain really promising.

5.2 Methodological assessment

The bibliographic survey in this thesis represented a significant part of the work because various fields of analysis were studied, such as metagenomics, subgroup analysis, deep learning, natural language processing or multiple instance learning. Another difficulty was to find an association between the two main types of machine learning algorithms studied, i.e Deep Learning and Subgroup Discovery, which have very different objectives.

All the proposed methods have been tested on various datasets with few examples, which does not allow for a unique train / validation / test split. We know that learning on these small bases can lead to overfitting phenomena especially when complex models are trained, and many statistical tests are performed. We have therefore carried out a large number of experiments, using cross-validation methods, to address this issue and allow the models to better generalize the data. Clearly, much experimentation (such as testing on other datasets) remains to be done to better account for the robustness of our approach.

An important point that we did not necessarily put forward at the beginning is that of simulation. Gradually, we saw its interest in training models and we sought to better understand the use of simulators. We found it relevant to establish simulation strategies to create metagenomic datasets. Indeeed, this allowed us to evaluate the performance of our models in classification and interpretability on different types of sequencing such as Illumina or Nanopore.

Finally, the development of our methods was hampered by resource requirements that were sometimes difficult to acquire. Indeed, the computing power required in CPU and GPU for data preprocessing and algorithm learning led to the use of several clusters (UMMISCO (local), MeSu (UPMC), Jolio-Curie (TGCC)). A significant

adaptation time of the code was essential to be able to deploy the models on each environment at our disposal.

5.3 Perspectives for future works

We summarize some areas of our proposed methods.

Deep learning models have demonstrated their learning power mainly on large datasets. In the field of metagenomics, the number of samples is only a few hundred while the total data is several terabytes. The models are complex, and it is difficult, with few samples, to fit them efficiently. To manage this issue, we can consider two approaches. Firstly, metagenomic data simulation software could be used with different simulation strategies acting as data augmentation techniques to train and improve our models. Secondly, there are families of neural networks designed to achieve good performances on small datasets, such as the *siamese network* [KZS15]. Future work includes exploring their use in classification or clustering of metagenome samples.

We used the *FastDNA* algorithm for the *Read2Genome* part but newer approaches could be considered in the pipeline, such as *Brume* [MV20], which handle larger k-mers. Moreover, the development of transformers, state-of-the-art models in natural language processing, could be an interesting architecture to be trained on DNA sequences although to our knowledge they have not yet obtained interesting results in this field.

Metagenome2Vec as promising results on raw Nanopore metagenomic data generated by simulators. However, it has not yet been tested on real Nanopore data sets. This is a valuable evaluation that need to be made as Nanopore sequencing represents the 3rd most used NGS technologies. In addition, this technology sequences reads in real time and could therefore be combined with streaming processing, in particular with the Spark Streaming framework [Zah+13], thus better enabling a “point-of-care” solution.

Due to the many variables contained in the metagenomic abundance tables, the subgroup discovery algorithms are trained on a large search space. *Q-Classifier* is impacted by this aspect and we have limited the complexity of the rules to 2 while the phenotype could be induced by more complex species interaction features. We plan to work on a rule generation process with a heuristic, such as beam search (like the one in predomics [Pri+20]), to reduce the search space and the number of

generated rules impacting statistical results. An interesting investigation would be to test our interpretable approach in concrete studies with fecal transplantation and analyze the changes involved.

As shown in a recent review on detection and prognosis of coronavirus (COVID-19) diseases [AIX+21], none of the machine learning models identified from 2,212 studies have clinical potential due to methodological flaws, including insufficient external validation. We recognize the importance of a robust validation process to ensure reproducibility and use of machine learning methodologies in the clinical domain. Thus, we plan to deploy our approaches in two nationally funded projects to achieve more reliable results and hopefully wider use of our methodology.

Bibliography

[Ada+16] Julien Adam, Tony Sourisseau, Ken A. Olaussen, et al. “MMS19 as a potential predictive marker of adjuvant chemotherapy benefit in resected non-small cell lung cancer”. In: *Cancer Biomarkers* 17.3 (Sept. 26, 2016), pp. 323–333 (cit. on p. 79).

[AS00] J Adolfsson and G Steineck. “Prognostic and treatment-predictive factors-is there a difference?” In: *Prostate cancer and prostatic diseases* 3.4 (2000), pp. 265–268 (cit. on p. 76).

[Ain20] Claire Ainsworth. *Therapeutic microbes to tackle disease*. Nature, 2020 (cit. on p. 22).

[AIX+21] AIX-COVNET, Michael Roberts, Derek Driggs, et al. “Common pitfalls and recommendations for using machine learning to detect and prognosticate for COVID-19 using chest radiographs and CT scans”. In: *Nature Machine Intelligence* 3.3 (Mar. 2021), pp. 199–217 (cit. on p. 123).

[Alm+21] Alexandre Almeida, Stephen Nayfach, Miguel Boland, et al. “A unified catalog of 204,938 reference genomes from the human gut microbiome”. In: *Nature Biotechnology* 39.1 (Jan. 2021), pp. 105–114 (cit. on p. 8).

[Alo+19] Md Zahangir Alom, Tarek M. Taha, Chris Yakopcic, et al. “A State-of-the-Art Survey on Deep Learning Theory and Architectures”. In: *Electronics* 8.3 (Mar. 5, 2019), p. 292 (cit. on p. 16).

[AAH19] Muaed Jamal Alomar, Khadeja Rashed Al-Ansari, and Najeeb A Hassan. “Comparison of awareness of diabetes mellitus type II with treatment’s outcome in term of direct cost in a hospital in Saudi Arabia”. In: *World Journal of Diabetes* 10.8 (Aug. 2019), pp. 463–472 (cit. on p. 93).

[Alv+20] A Alves, A Civet, A Laurent, et al. “Social deprivation aggravates post-operative morbidity in carcinologic colorectal surgery: Results of the COINCIDE multicenter study”. In: *Journal of Visceral Surgery* (July 2020), pp. 1–9 (cit. on p. 79).

[Ame17] American Diabetes Association. “6. Glycemic Targets”. In: *Diabetes Care* 40 (Supplement 1 Jan. 2017), S48–S56 (cit. on p. 97).

[Amr+15] Mourad Amrane, Alexandre Civet, Alexandre Templier, Dian Kang, and Francisco C Figueiredo. “Patients with Moderate to Severe Dry Eye Disease in Routine Clinical Practice in the UK - Physician and Patient’s Assessments”. In: *Investigative Ophthalmology Visual Science* 56.7 (2015), pp. 4443–4443 (cit. on p. 79).

[ALM17] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. “A simple but tough-to-beat baseline for sentence embeddings”. In: (2017), p. 16 (cit. on pp. 41, 50).

[Asc+20] Pablo Aschner, Juan J. Gagliardino, Hasan Ilkova, et al. “Persistent poor glycaemic control in individuals with type 2 diabetes in developing countries: 12 years of real-world evidence of the International Diabetes Management Practices Study (IDMPS)”. In: *Diabetologia* 63.4 (2020), pp. 711–721 (cit. on p. 34).

[Atz15] Martin Atzmueller. “Subgroup discovery”. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 5.1 (2015), pp. 35–49 (cit. on pp. 76, 80).

[Aud+17] Jérôme Audoux, Nicolas Philippe, Rayan Chikhi, et al. “DE-kupl: exhaustive capture of biological variation in RNA-seq data through k-mer decomposition”. In: *Genome Biology* 18.1 (Dec. 2017), p. 243 (cit. on p. 9).

[BCB16] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. “Neural Machine Translation by Jointly Learning to Align and Translate”. In: *arXiv:1409.0473 [cs, stat]* (May 19, 2016). arXiv: 1409.0473 (cit. on p. 16).

[Bak18] Amir Bakarov. “A Survey of Word Embeddings Evaluation Methods”. In: *arXiv:1801.09536 [cs]* (Jan. 21, 2018). arXiv: 1801.09536 (cit. on p. 47).

[Bal+18] Nicolás M Ballarini, Gerd K Rosenkranz, Thomas Jaki, Franz König, and Martin Posch. “Subgroup identification in clinical trials via the predicted individual treatment effect”. In: *PLoS ONE* 13.10 (2018), e0205971–22 (cit. on p. 76).

[BSR14] C Battioui, L Shen, and SJ Ruberg. “A resampling-based ensemble tree method to identify patient subgroups with enhanced treatment effect”. In: *Proceedings of the Joint Statistical Meetings*. 2014 (cit. on p. 75).

[BH14] Samy Bengio and Georg Heigold. “Word Embeddings for Speech Recognition”. In: (2014), p. 5 (cit. on p. 17).

[Ber+20] Gabriele Berg, Daria Rybakova, Doreen Fischer, et al. “Microbiome definition re-visited: old concepts and new challenges”. In: *Microbiome* 8.1 (Dec. 2020), p. 103 (cit. on p. 1).

[Bet19] Rebecca A. Betensky. “The *p* -Value Requires Context, Not a Threshold”. In: *The American Statistician* 73 (sup1 Mar. 29, 2019), pp. 115–117 (cit. on p. 87).

[Blu+18] Jeffrey D. Blume, Lucy D’Agostino McGowan, William D. Dupont, and Robert A. Greevy. “Second-generation p-values: Improved rigor, reproducibility, & transparency in statistical analyses”. In: *PLOS ONE* 13.3 (Mar. 22, 2018). Ed. by Neil R. Smalheiser, e0188299 (cit. on p. 87).

[Boj+16] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. “Enriching Word Vectors with Subword Information”. In: *arXiv:1607.04606 [cs]* (July 15, 2016). arXiv: 1607.04606 (cit. on p. 45).

[Bur+15] James F Burke, Jeremy B Sussman, David M Kent, and Rodney A Hayward. “Three simple rules to ensure reasonably credible subgroup analyses.” In: *BMJ* 351 (2015), h5651 (cit. on pp. 77, 83).

[Can+18] Toby P. Candler, Osama Mahmoud, Richard M. Lynn, et al. “Treatment adherence and BMI reduction are key predictors of HbA1c 1 year after diagnosis of childhood type 2 diabetes in the United Kingdom”. In: *Pediatric Diabetes* 19.8 (Dec. 2018), pp. 1393–1399 (cit. on p. 93).

[CPC19] Diogo V. Carvalho, Eduardo M. Pereira, and Jaime S. Cardoso. “Machine Learning Interpretability: A Survey on Methods and Metrics”. In: *Electronics* 8.8 (July 26, 2019), p. 832 (cit. on p. 15).

[Cha+20] Mark R. Charbonneau, Vincent M. Isabella, Ning Li, and Caroline B. Kurtz. “Developing a new class of engineered live bacterial therapeutics to treat human diseases”. In: *Nature Communications* 11.1 (Dec. 2020), p. 1738 (cit. on p. 2).

[CMZ12] Lei Chen, Dianna J. Magliano, and Paul Z. Zimmet. “The worldwide epidemiology of type 2 diabetes mellitus—present and future perspectives”. In: *Nature Reviews Endocrinology* 8.4 (Apr. 2012), pp. 228–236 (cit. on p. 34).

[Che+17] Shuai Chen, Lu Tian, Tianxi Cai, and Menggang Yu. “A general statistical framework for subgroup identification and comparative treatment scoring”. In: *Biometrics* 73.4 (2017), pp. 1199–1209 (cit. on p. 76).

[CM19] Charles Y. Chiu and Steven A. Miller. “Clinical metagenomics”. In: *Nature Reviews Genetics* 20.6 (June 2019), pp. 341–355 (cit. on p. 4).

[Con+17] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. “Supervised Learning of Universal Sentence Representations from Natural Language Inference Data”. In: *arXiv:1705.02364 [cs]* (May 5, 2017). arXiv: 1705.02364 (cit. on p. 50).

[Cri19] Alexis Criscuolo. “A fast alignment-free bioinformatics procedure to infer accurate distance-based phylogenetic trees from genome assemblies”. In: *Research Ideas and Outcomes* 5 (June 10, 2019), e36178 (cit. on p. 52).

[CCC16] Zhicheng Cui, Wenlin Chen, and Yixin Chen. “Multi-Scale Convolutional Neural Networks for Time Series Classification”. In: *arXiv:1603.06995 [cs]* (May 11, 2016). arXiv: 1603.06995 (cit. on p. 39).

[Dev+18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *arXiv:1810.04805 [cs]* (Oct. 10, 2018). arXiv: 1810.04805 (cit. on pp. 17, 50).

[Dhu+21] Eliza Dhungel, Yassin Mreyoud, Ho-Jin Gwak, et al. “MegaR: an interactive R package for rapid sample classification and phenotype prediction using metagenome profiles and machine learning”. In: *BMC Bioinformatics* 22.1 (Dec. 2021), p. 25 (cit. on p. 12).

[Dij+09] Bernadette Dijkman, Bauke Kooistra, Mohit Bhandari, and Evidence-Based Surgery Working Group. “How to work with a subgroup analysis.” In: *Canadian journal of surgery. Journal canadien de chirurgie* 52.6 (Dec. 2009), pp. 515–522 (cit. on pp. 78, 83–86, 92).

[DL14] Alex Dimitrienko and Ilya Lipkovitch. “SLIDES: Exploratory subgroup analysis: Post-hoc subgroup identification in clinical trials”. In: (2014), pp. 1–28 (cit. on p. 76).

[Doo+13] L L Doove, E Dusseldorp, K Van Deun, and I Van Mechelen. “A comparison of five recursive partitioning methods to find person subgroups involved in meaningful treatment–subgroup interactions”. In: *Advances in Data Analysis and Classification* 8.4 (2013), pp. 403–425 (cit. on p. 77).

[Dum+18] C Dumontet, C Hulin, MA Dimopoulos, et al. “A predictive model for risk of early grade3 infection in patients with multiple myeloma not eligible for transplant: analysis of the FIRST trial”. In: *Leukemia* 32.6 (2018), pp. 1404–1413 (cit. on p. 79).

[Dum+16] C Dumontet, C Hulin, MA Dimopoulos, et al. “Develpment of a predictive model to identify patients with multiple myeloma not eligible for autologous transplant at risk for severe infections using data from the first trial”. In: *Haematologica* 101 (2016) (cit. on p. 79).

[DCV10] Elise Dusseldorp, Claudio Conversano, and Bart Jan Van Os. “Combining an Additive and Tree-Based Regression Model Simultaneously: STIMA”. In: *Journal of Computational and Graphical Statistics* 19.3 (Jan. 2010), pp. 514–530 (cit. on p. 76).

[DDM16] Elise Dusseldorp, Lisa Doove, and Iven van Mechelen. “Quint: An R package for the identification of subgroups of clients who differ in which treatment alternative is best for them”. In: *Behavior Research Methods* (May 2016), pp. 1–14 (cit. on p. 76).

[EDF15] Edoardo Pasolli, Duy Tin Truong, and Faizan Malik. “Machine Learning Meta-analysis of Large Metagenomic Datasets: Tools and Biological Insights”. In: (2015) (cit. on pp. 12, 25).

[Esn+20] Cyril Esnault, May-Line Gadonna, Maxence Queyrel, Alexandre Templier, and Jean-Daniel Zucker. “Q-Finder: An Algorithm for Credible Subgroup Discovery in Clinical Data Analysis — An Application to the International Diabetes Management Practice Study”. In: *Frontiers in Artificial Intelligence* 3 (Dec. 17, 2020), p. 559927 (cit. on pp. 21, 71).

[Eve+14] C Eveno, Parc Y, Laurent A, et al. “An abnormal body mass index of is associated with an increased risk of rectosigmoid cancer risk: interest a short recto-sigmoidoscopy for early detection.” In: Vienna, Austria: United European Gastroenterology Journal, 2014 (cit. on p. 79).

[FTR11] Jared C Foster, Jeremy M G Taylor, and Stephen J Ruberg. “Subgroup identification from randomized clinical trial data”. In: *Statistics in Medicine* 30.24 (Aug. 2011), pp. 2867–2880 (cit. on pp. 21, 75).

[FA12] Jonathan Friedman and Eric J. Alm. “Inferring Correlation Networks from Genomic Survey Data”. In: *PLoS Computational Biology* 8.9 (Sept. 20, 2012). Ed. by Christian von Mering, e1002687 (cit. on pp. 14, 28, 29).

[Fri+19] Adrian Fritz, Peter Hofmann, Stephan Majda, et al. “CAMISIM: simulating metagenomes and microbial communities”. In: *Microbiome* 7.1 (Dec. 2019), p. 17 (cit. on pp. 6, 27, 54).

[FGL12] Johannes Fürnkranz, Dragan Gamberger, and Nada Lavrač. *Foundations of Rule Learning*. Springer, 2012, p. 353 (cit. on pp. 75, 76, 82).

[FGL14] Johannes Fürnkranz, Dragan Gamberger, and Nada Lavrač. *Foundations of Rule Learning*. Springer Publishing Company, Incorporated, 2014 (cit. on p. 20).

[Gan93] Jean-Gabriel Ganascia. “TDIS - an Algebraic Formalization.” In: *IJCAI* (1993) (cit. on p. 81).

[Gas+17] Y Gaston-Mathe, T Fan, A Shaunik, C Brulle-Wohlhueter, and Civet A. “Using machine learning algorithms to identify predictive factors of clinical outcomes with iGlarLixi or iGlar in the LixiLan-L trial”. In: *Diabetologia* 60.1 (2017), pp. 1–608 (cit. on p. 79).

[Geo+20] Andreas Georgiou, Vincent Fortuin, Harun Mustafa, and Gunnar Rätsch. “$META^2$: Memory-efficient taxonomic classification and abundance estimation for metagenomics with deep learning”. In: *arXiv:1909.13146 [cs, q-bio, stat]* (Feb. 10, 2020). arXiv: 1909.13146 (cit. on pp. 16, 42).

[Ges+19] Nils Gessert, Maximilian Nielsen, Mohsin Shaikh, René Werner, and Alexander Schlaefer. “Skin Lesion Classification Using Ensembles of Multi-Resolution EfficientNets with Meta Data”. In: *arXiv:1910.03910 [cs]* (Oct. 9, 2019). arXiv: 1910.03910 (cit. on p. 16).

[Gil+19] Leilani H. Gilpin, David Bau, Ben Z. Yuan, et al. “Explaining Explanations: An Overview of Interpretability of Machine Learning”. In: *arXiv:1806.00069 [cs, stat]* (Feb. 3, 2019). arXiv: 1806.00069 (cit. on p. 16).

[GP18] Geoffrey S. Ginsburg and Kathryn A. Phillips. “Precision Medicine: From Science To Value”. In: *Health Affairs* 37.5 (May 2018), pp. 694–701 (cit. on p. 3).

[GMM16] Sara Goodwin, John D. McPherson, and W. Richard McCombie. “Coming of age: ten years of next-generation sequencing technologies”. In: *Nature Reviews Genetics* 17.6 (June 2016), pp. 333–351 (cit. on p. 3).

[Gra+17] Edouard Grave, Armand Joulin, Moustapha Cissé, David Grangier, and Hervé Jégou. “Efficient softmax approximation for GPUs”. In: *arXiv:1609.04309 [cs]* (June 19, 2017). arXiv: 1609.04309 (cit. on p. 51).

[Hah+11] Michael Hahsler, Sudheer Chelluboina, Kurt Hornik, and Christian Buchta. “The arules R-Package Ecosystem: Analyzing Interesting Patterns from Large Transaction Data Sets”. In: *Journal of Machine Learning Research* 12 (June 2011), pp. 2021–2025 (cit. on p. 79).

[HHZ10] B. Hanczar, C. Henegar, and J. D. Zucker. “Exploring interaction measures to identify informative pairs of genes”. In: *Int J Bioinform Res Appl* 6.6 (2010), pp. 628–42 (cit. on pp. 77, 78).

[HB16a] Blaise Hanczar and Avner Bar-Hen. “Controlling the Cost of Prediction in using a Cascade of Reject Classifiers for Personalized Medicine:” in: *Proceedings of the 9th International Joint Conference on Biomedical Engineering Systems and Technologies*. 7th International Conference on Bioinformatics Models, Methods and Algorithms. Rome, Italy: SCITEPRESS - Science, 2016, pp. 42–50 (cit. on p. 23).

[HKH18] Alexander Hapfelmeier, Ulm Kurt, and Bernhard Haller. “Subgroup identification by recursive segmentation”. In: *Journal of Applied Statistics* 0.0 (Mar. 2018), pp. 1–24 (cit. on p. 76).

[Har+19] Zachary N. Harris, Eliza Dhungel, Matthew Mosior, and Tae-Hyuk Ahn. “Massive metagenomic data analysis using abundance-based machine learning”. In: *Biology Direct* 14.1 (Dec. 2019), p. 12 (cit. on pp. 9, 12).

[He+15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep Residual Learning for Image Recognition”. In: *arXiv:1512.03385 [cs]* (Dec. 10, 2015). arXiv: 1512.03385 (cit. on p. 39).

[HCR21] M. He, X. Chi, and J. Ren. “Applications of Oxford Nanopore Sequencing in Schizosaccharomyces pombe”. In: *Yeast Protocols*. Ed. by W Xiao. vol 2196. Humana, New York, NY: Methods in Molecular Biology, 2021 (cit. on p. 6).

[Hel16] Sumyea Helal. “Subgroup Discovery Algorithms: a Survey and Empirical Evaluation”. In: *Journal of Computer Science and Technology* 31.3 (2016), pp. 561–576 (cit. on pp. 76, 77, 82).

[Her+10] Franciso Herrera, Cristóbal José Carmona, Pedro González, and Maria José del Jesus. “An overview on subgroup discovery: foundations and applications”. In: *Knowledge and Information Systems* 29.3 (2010), pp. 495–525 (cit. on pp. 77, 79).

[HB16b] Seth Hildick-Smith and Ivaylo Bahtchevanov. “Deep Learning for Natural Language Sequence Labelling Applied to Epigenomics”. In: (2016), p. 8 (cit. on p. 39).

[HCK16] Felix Hill, Kyunghyun Cho, and Anna Korhonen. “Learning Distributed Representations of Sentences from Unlabelled Data”. In: *arXiv:1602.03483 [cs]* (Feb. 10, 2016). arXiv: 1602.03483 (cit. on pp. 17, 50).

[Hol16] Andreas Holzinger. “Interactive machine learning for health informatics: when do we need the human-in-the-loop?” In: *Brain Informatics* 3.2 (Feb. 2016), pp. 119–131 (cit. on p. 92).

[Hua+12] Weichun Huang, Leping Li, Jason R. Myers, and Gabor T. Marth. “ART: a next-generation sequencing read simulator”. In: *Bioinformatics* 28.4 (Feb. 15, 2012), pp. 593–594 (cit. on p. 4).

[HY18] Jared D Huling and Menggang Yu. “Subgroup Identification Using the personalized Package”. In: *arXiv.org* (2018) (cit. on p. 76).

[ITW18] Maximilian Ilse, Jakub M. Tomczak, and Max Welling. “Attention-based Deep Multiple Instance Learning”. In: *arXiv:1802.04712 [cs, stat]* (Feb. 13, 2018). arXiv: 1802.04712 (cit. on pp. 59, 61, 151).

[IR13] Kosuke Imai and Marc Ratkovic. “Estimating treatment effect heterogeneity in randomized program evaluation”. In: *The Annals of Applied Statistics* 7.1 (Mar. 2013), pp. 443–470 (cit. on p. 75).

[Imp12] Andrea Imparato. “Interactive Subgroup Discovery”. In: (2012), p. 134 (cit. on p. 20).

[Jai+16] Miten Jain, Hugh E. Olsen, Benedict Paten, and Mark Akeson. “The Oxford Nanopore MinION: delivery of nanopore sequencing to the genomics community”. In: *Genome Biology* 17.1 (Dec. 2016), p. 239 (cit. on p. 6).

[Job18] Christian Jobin. “Precision medicine using microbiota”. In: *Science* 359.6371 (Jan. 5, 2018), pp. 32–34 (cit. on pp. 3, 37).

[Jou+16] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. “Bag of Tricks for Efficient Text Classification”. In: *arXiv:1607.01759 [cs]* (July 6, 2016). arXiv: 1607.01759 (cit. on pp. 17, 41, 45, 49).

[Jua+12] Deborah Taira Juarez, Tetine Sentell, Sheri Tokumaru, et al. “Factors Associated With Poor Glycemic Control or Wide Glycemic Variability Among Diabetes Patients in Hawaii, 2006–2009”. In: *Preventing Chronic Disease* 9 (Sept. 27, 2012), p. 120065 (cit. on p. 93).

[KL07] Branko Kavsek and Nada Lavrač. “APRIORI-SD: Adapting Association Rule Learning to Subgroup Discovery”. In: *Applied Artificial Intelligence* 20.7 (2007), pp. 543–583 (cit. on p. 21).

[KSR16] David R Kelley, Jasper Snoek, and John L Rinn. “Basset: Learning the regulatory code of the accessible genome with deep convolutional neural networks”. In: (2016), p. 35 (cit. on p. 39).

[KW14] Diederik P. Kingma and Max Welling. “Auto-Encoding Variational Bayes”. In: *arXiv:1312.6114 [cs, stat]* (May 1, 2014). arXiv: 1312.6114 (cit. on pp. 17, 18, 42, 59).

[Kir+15] Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, et al. “Skip-Thought Vectors”. In: *arXiv:1506.06726 [cs]* (June 22, 2015). arXiv: 1506.06726 (cit. on p. 50).

[KZS15] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. “Siamese Neural Networks for One-shot Image Recognition”. In: (2015), p. 8 (cit. on p. 122).

[Kor+21] Krzysztof Koras, Ewa Kizling, Dilafruz Juraeva, Eike Staub, and Ewa Szczurek. *Interpretable deep recommender system model for prediction of kinase inhibitor efficacy across cancer cell lines*. preprint. Bioinformatics, Jan. 27, 2021 (cit. on p. 16).

[Kor18] Natalia V Korepanova. “Subgroup Discovery for Treatment Optimization”. In: (2018), pp. 1–6 (cit. on p. 3).

[Kun+08] V. Kunin, A. Copeland, A. Lapidus, K. Mavromatis, and P. Hugenholtz. “A Bioinformatician’s Guide to Metagenomics”. In: *Microbiology and Molecular Biology Reviews* 72.4 (Dec. 1, 2008), pp. 557–578 (cit. on p. 7).

[Lav+04] Nada Lavrač, Branko Kavsek, Peter A Flach, and Ljupco Todorovski. “Subgroup Discovery with CN2-SD.” In: *Journal of Machine Learning Research* (2004) (cit. on pp. 21, 76).

[Lee+21] Emily R. Leeming, Panayiotis Louca, Rachel Gibson, et al. “The complexities of the diet-microbiome relationship: advances and perspectives”. In: *Genome Medicine* 13.1 (Dec. 2021), p. 10 (cit. on p. 4).

[LK12] Matthijs van Leeuwen and Arno Knobbe. “Diverse subgroup set discovery”. In: *Data Mining and Knowledge Discovery* 25.2 (2012), pp. 208–242 (cit. on p. 89).

[LB18] Florian Lemmerich and Martin Becker. “pysubgroup: Easy-to-use subgroup discovery in python”. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. 2018, pp. 658–662 (cit. on p. 95).

[LB13] R. Levy and E. Borenstein. “Metabolic modeling of species interaction in the human microbiome elucidates community-level assembly rules”. In: *Proceedings of the National Academy of Sciences* 110.31 (July 30, 2013), pp. 12804–12809 (cit. on p. 29).

[Li+03] J. Li, H. Liu, J. R. Downing, A. E.-J. Yeoh, and L. Wong. “Simple rules underlying gene expression profiles of more than six subtypes of acute lymphoblastic leukemia (ALL) patients”. In: *Bioinformatics* 19.1 (Jan. 1, 2003), pp. 71–78 (cit. on p. 20).

[Lia+18] Dachao Liang, Ross Ka-Kit Leung, Wenda Guan, and William W. Au. “Involvement of gut microbiome in human health and disease: brief overview, knowledge gaps and research opportunities”. In: *Gut Pathogens* 10.1 (Dec. 2018), p. 3 (cit. on p. 3).

[Lia+20] Qiaoxing Liang, Paul W Bible, Yu Liu, Bin Zou, and Lai Wei. “DeepMicrobes: taxonomic classification for metagenomics with deep learning”. In: *NAR Genomics and Bioinformatics* 2.1 (Mar. 1, 2020), lqaa009 (cit. on pp. 16, 38, 41, 55).

[Lip+16] Ilya Lipkovich et al. “Tutorial in biostatistics: data-driven subgroup identification and analysis in clinical trials”. In: *Statistics in Medicine* 36.1 (2016), pp. 136–196 (cit. on pp. 72–74, 76).

[LD14] Ilya Lipkovich and Alex Dmitrienko. “Strategies for Identifying Predictive Biomarkers and Subgroups with Enhanced Treatment Effect in Clinical Trials Using SIDES”. In: *Journal of Biopharmaceutical Statistics* 24.1 (2014), pp. 130–153 (cit. on pp. 21, 75).

[Lip+18] Ilya Lipkovich, Alex Dmitrienko, Christoph Muysers, and Bohdana Ratitch. “Multiplicity issues in exploratory subgroup analysis”. In: *Journal of Biopharmaceutical Statistics* 28.1 (2018), pp. 63–81 (cit. on pp. 74, 76).

[Lip+17] Ilya Lipkovich, Alex Dmitrienko, Kaushik Patra, Bohdana Ratitch, and Erik Pulkstenis. “Subgroup Identification in Clinical Trials by Stochastic SIDEScreen Methods”. In: *Statistics in Biopharmaceutical Research* 9.4 (2017), pp. 368–378 (cit. on p. 76).

[LCZ19] Wei-Yin Loh, Luxi Cao, and Peigen Zhou. “Subgroup identification for precision medicine: A comparative review of 13 methods”. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 9.5 (2019), pp. 604–21 (cit. on pp. 3, 77).

[Lor+20] Claudio Lorenzi, Sylvain Barriere, Jean-Philippe Villemin, et al. “iMOKA: k-mer based software to analyze large collections of sequencing data”. In: *Genome Biology* 21.1 (Dec. 2020), p. 261 (cit. on p. 13).

[LL17] Scott M Lundberg and Su-In Lee. “A Unified Approach to Interpreting Model Predictions”. In: (2017), p. 10 (cit. on pp. 111, 117).

[LPM15] Thang Luong, Hieu Pham, and Christopher D. Manning. “Effective Approaches to Attention-based Neural Machine Translation”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon, Portugal: Association for Computational Linguistics, 2015, pp. 1412–1421 (cit. on pp. 16, 50, 151).

[Ma+16] Qinglin Ma, Houming Liu, Guangxin Xiang, Wanshui Shan, and Wanli Xing. “Association between glycated hemoglobin A1c levels with age and gender in Chinese adults with no prior diagnosis of diabetes mellitus”. In: *Biomedical Reports* 4.6 (2016), pp. 737–740 (cit. on p. 93).

[Maa+11] Andrew L Maas, Raymond E Daly, Peter T Pham, et al. “Learning Word Vectors for Sentiment Analysis”. In: (2011), p. 9 (cit. on p. 17).

[Mar17] Elaine R Mardis. “DNA sequencing technologies: 2006–2016”. In: *Nature Protocols* 12.2 (Jan. 5, 2017), pp. 213–218 (cit. on p. 3).

[MLD15] Cristiana Mayer, Ilya Lipkovich, and Alex Dmitrienko. “Survey Results on Industry Practices and Challenges in Subgroup Analysis in Clinical Trials”. In: *Statistics in Biopharmaceutical Research* 7.4 (2015), pp. 272–282 (cit. on p. 77).

[MHM20] Leland McInnes, John Healy, and James Melville. “UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction”. In: *arXiv:1802.03426 [cs, stat]* (Sept. 17, 2020). arXiv: 1802.03426 (cit. on p. 65).

[MV19] Romain Menegaux and Jean-Philippe Vert. “Continuous Embeddings of DNA Sequencing Reads and Application to Metagenomics”. In: *Journal of Computational Biology* 26.6 (June 2019), pp. 509–518 (cit. on pp. 16, 17, 38, 40, 54, 55, 63, 109).

[MV20] Romain Menegaux and Jean-Philippe Vert. *Embedding the de Bruijn graph, and applications to metagenomics*. preprint. Bioinformatics, Mar. 8, 2020 (cit. on p. 122).

[Met+13] MetaHIT consortium, Emmanuelle Le Chatelier, Trine Nielsen, et al. “Richness of human gut microbiome correlates with metabolic markers”. In: *Nature* 500.7464 (Aug. 2013), pp. 541–546 (cit. on p. 25).

[Met+14a] MetaHIT Consortium, Junhua Li, Huijue Jia, et al. “An integrated catalog of reference genes in the human gut microbiome”. In: *Nature Biotechnology* 32.8 (Aug. 2014), pp. 834–841 (cit. on pp. 8, 27).

[Met+14b] MetaHIT Consortium, H Bjørn Nielsen, Mathieu Almeida, et al. “Identification and assembly of genomes and genetic elements in complex metagenomic samples without using reference genomes”. In: *Nature Biotechnology* 32.8 (Aug. 2014), pp. 822–828 (cit. on pp. 7–9).

[Mik+13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. “Distributed Representations of Words and Phrases and their Compositionality”. In: (2013), p. 9 (cit. on pp. 17, 41, 45, 49).

[Min+17] Xu Min, Wanwen Zeng, Ning Chen, Ting Chen, and Rui Jiang. “Chromatin accessibility prediction via convolutional long short-term memory networks with k-mer embedding”. In: *Bioinformatics* 33.14 (July 15, 2017), pp. i92–i101 (cit. on pp. 16, 38, 40, 48).

[Mol21] Christophe Molnar. *Interpretable Machine Learning*. 2021 (cit. on pp. 16, 20).

[Moo01] Dwayne R.J. Moore. “The *Anna Karenina* Principle Applied to Ecological Risk Assessments of Multiple Stressors”. In: *Human and Ecological Risk Assessment: An International Journal* 7.2 (Mar. 2001), pp. 231–237 (cit. on p. 19).

[Mor+20] Etienne Mornet, Rollot Mélissa, Hlioui Cyrine, et al. “Recherche de SNP modulateurs du phénotype hypophosphatasique par un algorithme d’identification de règles d’association (« subgroup discovery »)”. In: Tours, France: Assises de Génétique Humaine et Médicale, 2020 (cit. on p. 79).

[Nab+12] J-M Nabholtz, M-M Dauplat, C Abrial, et al. “Abstract P3-06-20: Is it possible to predict the efficacy of a combination of Panitumumab plus FEC 100 followed by docetaxel (T) for patients with triple negative breast cancer (TNBC)? Final biomarker results from a phase II neoadjuvant trial.” In: *Cancer Research* 72.24 Supplement (2012), P3-06-20–P3-06-20. eprint: https://cancerres.aacrjournals.org/content/72/24_Supplement/P3-06-20 (cit. on p. 79).

[NF18] Hidewaki Nakagawa and Masashi Fujita. “Whole genome sequencing analysis for cancer genomics and precision medicine”. In: *Cancer Science* 109.3 (Mar. 2018), pp. 513–522 (cit. on p. 4).

[NP16] Stephen Nayfach and Katherine S. Pollard. “Toward Accurate and Quantitative Comparative Metagenomics”. In: *Cell* 166.5 (Aug. 2016), pp. 1103–1116 (cit. on p. 7).

[NW70] Saul B. Needleman and Christian D. Wunsch. “A general method applicable to the search for similarities in the amino acid sequence of two proteins”. In: *Journal of Molecular Biology* 48.3 (1970), pp. 443–453 (cit. on p. 48).

[Ng17] Patrick Ng. “dna2vec: Consistent vector representations of variable-length k-mers”. In: *arXiv:1701.06279 [cs, q-bio, stat]* (Jan. 23, 2017). arXiv: 1701.06279 (cit. on pp. 17, 40, 47).

[Ngu+] Ha Nguyen, Haoliang Xue, Virginie Firlej, et al. “A Comparative Analysis of Reference-Free and Conventional Transcriptome Signatures for Prostate Cancer Prognosis”. In: (), p. 28 (cit. on p. 13).

[NLW09] Petra Kralj Novak, Nada Lavrac˘, and Geoffrey I Webb. “Supervised Descriptive Rule Discovery: A Unifying Survey of Contrast Set, Emerging Pattern and Subgroup Mining”. In: (2009), p. 27 (cit. on p. 20).

[Ogu+17] K Ogurtsova, J D da Rocha Fernandes, Y HUANG, et al. “IDF Diabetes Atlas: Global estimates for the prevalence of diabetes for 2015 and 2040.” In: *Diabetes research and clinical practice* 128 (June 2017), pp. 40–50 (cit. on p. 34).

[OZ20] Min Oh and Liqing Zhang. “DeepMicro: deep representation learning for disease prediction based on microbiome data”. In: *Scientific Reports* 10.1 (Dec. 2020), p. 6026 (cit. on pp. 12, 62, 63, 120).

[OC15] Anastasis Oulas and Christina Pavloudi. “Metagenomics: Tools and Insights for Analyzing Next-Generation Sequencing Data Derived from Biodiversity Studies”. In: (2015) (cit. on p. 3).

[OG92] A D Oxman and G H Guyatt. “A consumer’s guide to subgroup analyses.” In: *Annals of Internal Medicine* 116.1 (Jan. 1992), pp. 78–84 (cit. on p. 83).

[Pas+16] Edoardo Pasolli, Duy Tin Truong, Faizan Malik, Levi Waldron, and Nicola Segata. “Machine Learning Meta-analysis of Large Metagenomic Datasets: Tools and Biological Insights”. In: *PLOS Computational Biology* 12.7 (July 11, 2016). Ed. by Jonathan A. Eisen, e1004977 (cit. on pp. 12, 23, 25, 26, 62, 63, 66, 109, 120).

[Pas+19] Adam Paszke, Sam Gross, Francisco Massa, et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *arXiv:1912.01703 [cs, stat]* (Dec. 3, 2019). arXiv: 1912.01703 (cit. on p. 35).

[PPV19] Leonardo Pellegrina, Cinzia Pizzi, and Fabio Vandin. “Fast Approximation of Frequent k-mers and Applications to Metagenomics”. In: *arXiv:1902.10168 [q-bio]* (Feb. 26, 2019). arXiv: 1902.10168 (cit. on p. 9).

[PSM14] Jeffrey Pennington, Richard Socher, and Christopher Manning. “Glove: Global Vectors for Word Representation”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar: Association for Computational Linguistics, 2014, pp. 1532–1543 (cit. on pp. 40, 46, 49).

[PGB20] Ana Elena Pérez-Cobas, Laura Gomez-Valero, and Carmen Buchrieser. “Metagenomic approaches in microbial ecology: an update on whole-genome and marker gene sequencing analyses”. In: *Microbial Genomics* 6.8 (Aug. 1, 2020) (cit. on pp. 10, 119).

[Pet18] Joseph F. Petrosino. “The microbiome in precision medicine: the way forward”. In: *Genome Medicine* 10.1 (Dec. 2018), p. 12 (cit. on p. 1).

[Pet+18] Goran Petrovski, Dashamir Gjergji, Aleksandra Grbic, et al. “Switching From Pre-mixed Insulin to Regimens with Insulin Glargine in Type 2 Diabetes: A Prospective, Observational Study of Data From Adriatic Countries”. In: *Diabetes Therapy* 9.4 (Aug. 2018), pp. 1657–1668 (cit. on p. 93).

[Pla+19] Florian Plaza Oñate, Emmanuelle Le Chatelier, Mathieu Almeida, et al. “MSP-miner: abundance-based reconstitution of microbial pan-genomes from shotgun metagenomic data”. In: *Bioinformatics* 35.9 (May 1, 2019). Ed. by Jonathan Wren, pp. 1544–1552 (cit. on p. 8).

[PW10] Wolfgang Polonik and Zailong Wang. “PRIM analysis”. In: *Journal of Multivariate Analysis* 101.3 (Mar. 2010), pp. 525–540 (cit. on p. 75).

[Pri+20] Edi Prifti, Yann Chevaleyre, Blaise Hanczar, et al. “Interpretable and accurate prediction models for metagenomics data”. In: *GigaScience* 9.3 (Mar. 1, 2020), giaa010 (cit. on pp. 12, 15, 25, 109, 112, 122).

[Qi+18] Ye Qi, Devendra Singh Sachan, Matthieu Felix, Sarguna Janani Padmanabhan, and Graham Neubig. “When and Why are Pre-trained Word Embeddings Useful for Neural Machine Translation?” In: *arXiv:1804.06323 [cs]* (Apr. 18, 2018). arXiv: 1804.06323 (cit. on p. 17).

[QC19] Jia Qian and Matteo Comin. “MetaCon: unsupervised clustering of metagenomic contigs with probabilistic k-mers statistics and coverage”. In: *BMC Bioinformatics* 20 (S9 Nov. 2019), p. 367 (cit. on p. 7).

[Qin12] Junjie Qin. “A metagenome-wide association study of gut microbiota in type 2 diabetes”. In: (2012), p. 6 (cit. on p. 25).

[Qin+14] Nan Qin, Fengling Yang, Ang Li, et al. “Alterations of the human gut microbiome in liver cirrhosis”. In: *Nature* 513.7516 (Sept. 4, 2014), pp. 59–64 (cit. on pp. 25, 29, 33).

[QX16] Daniel Quang and Xiaohui Xie. “DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences”. In: *Nucleic Acids Research* 44.11 (June 20, 2016), e107–e107 (cit. on p. 39).

[Que+21] Maxence Queyrel, Edi Prifti, Alexandre Templier, and Jean-Daniel Zucker. “Towards End-To-End Disease Prediction from Raw Metagenomic Data”. In: *International Journal of Biomedical and Biological Engineering* 15.6 (2021), pp. 234–246 (cit. on p. 18).

[Qui+17] Christopher Quince, Alan W Walker, Jared T Simpson, Nicholas J Loman, and Nicola Segata. “Shotgun metagenomics, from sampling to sequencing and analysis”. In: (2017), p. 27 (cit. on pp. 7, 12).

[QE20] Thomas P. Quinn and Ionas Erb. “Interpretable Log Contrasts for the Classification of Health Biomarkers: a New Approach to Balance Selection”. In: *mSystems* 5.2 (Apr. 7, 2020). Ed. by Robert G. Beiko, e00230–19, /msystems/5/2/msys.00230–19.atom (cit. on p. 14).

[RR18] Mohammad Arifur Rahman and Huzefa Rangwala. “RegMIL: Phenotype Classification from Metagenomic Data”. In: *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics - BCB ’18*. the 2018 ACM International Conference. Washington, DC, USA: ACM Press, 2018, pp. 145–154 (cit. on p. 41).

[Rib+20] Antônio H. Ribeiro, Manoel Horta Ribeiro, Gabriela M. M. Paixão, et al. “Automatic diagnosis of the 12-lead ECG using a deep neural network”. In: *Nature Communications* 11.1 (Dec. 2020), p. 1760. arXiv: 1904.01949 (cit. on p. 16).

[Roj+19] Mateo Rojas-Carulla, Ilya Tolstikhin, Guillermo Luque, et al. “GeNet: Deep Representations for Metagenomics”. In: *arXiv:1901.11015 [cs, q-bio, stat]* (Jan. 30, 2019). arXiv: 1901.11015 (cit. on pp. 16, 39).

[Rol+18] M. Rollot, M. Bonnemaire, C. Brulle-Wohlhueter keywords=main, et al. “A machine learning algorithm can identify clusters of patients with favourable glycaemic outcomes in a pooled European Gla-300 studies (REALI): Novel signposts for clinicians?” In: *Diabetologia : journal of the European Association for the Study of Diabetes (EASD)* 61.Supplement 1 (Oct. 1, 2018), p. 876 (cit. on p. 79).

[Rot05] Peter M Rothwell. “Subgroup analysis in randomised controlled trials: importance, indications, and interpretation”. In: *The Lancet* 365.9454 (2005), pp. 176–186 (cit. on pp. 73, 78).

[Rüc+18] Andreas Rücklé, Steffen Eger, Maxime Peyrard, and Iryna Gurevych. “Concatenated Power Mean Word Embeddings as Universal Cross-Lingual Sentence Representations”. In: *arXiv:1803.01400 [cs]* (Mar. 4, 2018). arXiv: 1803.01400 (cit. on p. 50).

[Sam+20] Wojciech Samek, Grégoire Montavon, Sebastian Lapuschkin, Christopher J. Anders, and Klaus-Robert Müller. “Toward Interpretable Machine Learning: Transparent Deep Neural Networks and Beyond”. In: *arXiv:2003.07631 [cs, stat]* (Mar. 17, 2020). arXiv: 2003.07631 (cit. on p. 16).

[Sat+14] S Saturni, F Bellini, F Braido, et al. “Randomized controlled trials and real life studies. Approaches and methodologies: a clinical point of view.” In: *Pulmonary Pharmacology amp; Therapeutics* 27.2 (2014), pp. 129–138 (cit. on p. 72).

[Sch+16] Patrick M Schnell, Qi Tang, Walter W Offen, and Bradley P Carlin. “A Bayesian credible subgroups approach to identifying patient subgroups with positive treatment effects”. In: *Biometrics* 72.4 (2016), pp. 1026–1036 (cit. on p. 83).

[Sco+15] Karen P. Scott, Jean-Michel Antoine, Tore Midtvedt, and Saskia van Hemert. “Manipulating the gut microbiota to maintain health and treat disease”. In: *Microbial Ecology in Health & Disease* 26.0 (Feb. 2, 2015) (cit. on p. 22).

[Seg+11] Nicola Segata, Jacques Izard, Levi Waldron, et al. “Metagenomic biomarker discovery and explanation”. In: *Genome Biology* 12.6 (2011), R60 (cit. on p. 12).

[SM16] Ron Sender and Ron Milo. “Revised estimates for the number of human and bacteria cells in the body”. In: (2016), p. 21 (cit. on p. 2).

[SPW16] Grace Tzun-Wen Shaw, Yueh-Yang Pao, and Daryi Wang. “MetaMIS: a metagenomic microbial interaction simulator based on microbial community profiles”. In: *BMC Bioinformatics* 17.1 (Dec. 2016), p. 488 (cit. on p. 29).

[SSZ10] J.E. Shaw, R.A. Sicree, and P.Z. Zimmet. “Global estimates of the prevalence of diabetes for 2010 and 2030”. In: *Diabetes Research and Clinical Practice* 87.1 (Jan. 2010), pp. 4–14 (cit. on p. 34).

[SJF14] Julius Stecher, Frederik Janssen, and Johannes Fürnkranz. “Separating Rule Refinement and Rule Selection Heuristics in Inductive Rule Learning”. In: *Machine Learning and Knowledge Discovery in Databases*. Ed. by Toon Calders, Floriana Esposito, Eyke Hüllermeier, and Rosa Meo. Vol. 8726. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 114–129 (cit. on p. 21).

[Su+09] Xiaogang Su, Chih-Ling Tsai, Hansheng Wang, David M. Nickerson, and Bogong Li. “Subgroup Analysis via Recursive Partitioning”. In: *SSRN Electronic Journal* (2009) (cit. on p. 75).

[SBJ12] Xin Sun, Matthias Briel, and Busse Jason. “Credibility of claims of subgroup effects in randomised controlled trials: systematic review”. In: *BMJ* 344.mar15 1 (2012), e1553–e1553 (cit. on p. 83).

[Sun+10] Xin Sun, Matthias Briel, Stephen Walter, and Gordon Guyatt. “Is a subgroup effect believable? Updating criteria to evaluate the credibility of subgroup analyses”. In: *BMJ* 340.mar30 3 (2010), pp. c117–c117 (cit. on pp. 78, 83–85, 87, 92).

[Sun+14] Xin Sun, John Ioannidis, Thomas Agoritsas, Ana Alba, and Gordon Guyatt. “How to Use a Subgroup Analysis”. In: *JAMA* 311.4 (2014), pp. 405–7 (cit. on p. 74).

[SKP10] RPh Susan L Dennett, RPh Kristina S Boye, and Nicole R Yurgin PhD. “The Impact of Body Weight on Patient Utilities with or without Type 2 Diabetes: A Review of the Medical Literature”. In: *Value in Health* 11.3 (Oct. 2010), pp. 478–486 (cit. on p. 93).

[SVL14] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. “Sequence to Sequence Learning with Neural Networks”. In: *arXiv:1409.3215 [cs]* (Sept. 10, 2014). arXiv: 1409.3215 (cit. on pp. 17, 49).

[Tan+16] Julien Tanniou, Ingeborg van der Tweel, Steven Teerenstra, and Kit C B Roes. “Subgroup analyses in confirmatory clinical trials: time to be specific about their purposes”. In: *BMC Medical Research Methodology* (2016), pp. 1–15 (cit. on p. 72).

[Tho+19] Andrew Maltez Thomas, Paolo Manghi, Francesco Asnicar, et al. “Metagenomic analysis of colorectal cancer datasets identifies cross-cohort microbial diagnostic signatures and a link with choline degradation”. In: *Nature Medicine* 25.4 (Apr. 2019), pp. 667–678 (cit. on pp. 12, 25).

[Tru+15] Duy Tin Truong, Eric A Franzosa, Timothy L Tickle, et al. “MetaPhlAn2 for enhanced metagenomic taxonomic profiling”. In: *Nature Methods* 6.8 (2015) (cit. on p. 26).

[Tsh+12] Jacques Kande Tshiang Tshiananga, Serge Kocher, Christian Weber, et al. “The Effect of Nurse-led Diabetes Self-management Education on Glycosylated Hemoglobin and Cardiovascular Risk Factors: A Meta-analysis”. In: *The Diabetes Educator* 38.1 (Jan. 2012), pp. 108–123 (cit. on p. 93).

[Uga+19] Ari Ugarte, Minh Quang-Dao, Bich Hai Ho, et al. “QMSpy: An Integrated Modular and Scalable Platform for Quantitative Metagenomics in Pyspark”. In: *2019 IEEE-RIVF International Conference on Computing and Communication Technologies (RIVF)*. 2019 IEEE-RIVF International Conference on Computing and Communication Technologies (RIVF). Danang, Vietnam: IEEE, Mar. 2019, pp. 1–6 (cit. on p. 14).

[Val+17a] Anita Valmarska, Nada Lavrač, Johannes Fürnkranz, and Marko Robnik-Šikonja. “Refinement and selection heuristics in subgroup discovery and classification rule learning”. In: (2017), pp. 1–16 (cit. on p. 77).

[Val+17b] Anita Valmarska, Nada Lavrač, Johannes Fürnkranz, and Marko Robnik-Šikonja. “Refinement and selection heuristics in subgroup discovery and classification rule learning”. In: *Expert Systems with Applications* 81 (Sept. 2017), pp. 147–162 (cit. on pp. 20, 21).

[WZ00] Jun Wang and Jean-Daniel Zucker. *Solving Multiple-Instance Problem: a Lazy Learning Approach*. 2000 (cit. on p. 41).

[WSL19] Ronald L. Wasserstein, Allen L. Schirm, and Nicole A. Lazar. “Moving to a World Beyond “$p < 0.05$””. In: *The American Statistician* 73.sup1 (2019), pp. 1–19. eprint: https://doi.org/10.1080/00031305.2019.1583913 (cit. on pp. 86, 92).

[Wei+16] Sophie Weiss, Will Van Treuren, Catherine Lozupone, et al. “Correlation detection strategies in microbial data sets vary widely in sensitivity and precision”. In: *The ISME Journal* 10.7 (July 2016), pp. 1669–1681 (cit. on p. 29).

[Wen+17] Chengping Wen, Zhijun Zheng, Tiejuan Shao, et al. “Quantitative metagenomics reveals unique gut microbiome biomarkers in ankylosing spondylitis”. In: *Genome Biology* 18.1 (Dec. 2017), p. 142 (cit. on p. 7).

[Wet20] Kris Wetterstrand. *DNA Sequencing Costs: Data from the NHGRI Genome Sequencing Program (GSP)*. https://www.genome.gov/about-genomics/fact-sheets/DNA-Sequencing-Costs-Data. Accessed: 2021-03-11. 2020 (cit. on p. 3).

[Wij+17] Rients P. T. van Wijngaarden, Jetty A. Overbeek, Edith M. Heintjes, et al. “Relation Between Different Measures of Glycemic Exposure and Microvascular and Macrovascular Complications in Patients with Type 2 Diabetes Mellitus: An Observational Cohort Study”. In: *Diabetes Therapy* 8.5 (2017), pp. 1097–1109 (cit. on p. 93).

[Wir+19] Jakob Wirbel, Paul Theodor Pyl, Ece Kartal, et al. “Meta-analysis of fecal metagenomes reveals global microbial signatures that are specific for colorectal cancer”. In: *Nature Medicine* 25.4 (Apr. 2019), pp. 679–689 (cit. on p. 3).

[Wol+18] Stephen Woloszynek, Zhengqiao Zhao, Jian Chen, and Gail L. Rosen. “16S rRNA sequence embeddings: Meaningful numeric feature representations of nucleotide sequences that are convenient for downstream analyses”. In: *bioRxiv* (May 4, 2018) (cit. on pp. 16, 38, 41).

[Wu+21] Guojun Wu, Naisi Zhao, Chenhong Zhang, Yan Y. Lam, and Liping Zhao. “Guild-based analysis for understanding gut microbiome in human health and diseases”. In: *Genome Medicine* 13.1 (Dec. 2021), p. 22 (cit. on p. 3).

[XBM06] Hui Xiong, Mark Brodie, and Sheng Ma. “TOP-COP - Mining TOP-K Strongly Correlated Pairs in Large Databases.” In: *ICDM* (2006), pp. 1162–1166 (cit. on p. 89).

[Xu+15] Yaoyao Xu, Menggang Yu, Ying-Qi Zhao, et al. “Regularized outcome weighted subgroup identification for differential treatment effects”. In: *Biometrics* 71.3 (2015), pp. 645–653 (cit. on p. 76).

[Yan+17] Chen Yang, Justin Chu, Rene L Warren, and Inanc Birol. “NanoSim: nanopore sequence read simulator based on statistical characterization”. In: (2017), p. 6 (cit. on pp. 6, 27, 114).

[YZG21] Fenglong Yang, Quan Zou, and Bo Gao. “GutBalance: a server for the human gut microbiome-based disease prediction and biomarker discovery with compositionality addressed”. In: *Briefings in Bioinformatics* (Jan. 30, 2021), bbaa436 (cit. on pp. 12–15).

[You+18] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. “Recent Trends in Deep Learning Based Natural Language Processing”. In: *arXiv:1708.02709 [cs]* (Nov. 24, 2018). arXiv: 1708.02709 (cit. on p. 39).

[Zah+12] Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, et al. “Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing”. In: (2012), p. 14 (cit. on p. 35).

[Zah+13] Matei Zaharia, Tathagata Das, Haoyuan Li, et al. “Discretized streams: fault-tolerant streaming computation at scale”. In: *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles*. SOSP ’13: ACM SIGOPS 24th Symposium on Operating Systems Principles. Farminton Pennsylvania: ACM, Nov. 3, 2013, pp. 423–438 (cit. on p. 122).

[Zah+17] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, et al. “Deep Sets”. In: *arXiv:1703.06114 [cs, stat]* (Mar. 10, 2017). arXiv: 1703.06114 (cit. on pp. 18, 42, 59, 61, 151).

[ZMV17] Jesse R. Zaneveld, Ryan McMinds, and Rebecca Vega Thurber. “Stress and stability: applying the Anna Karenina principle to animal microbiomes”. In: *Nature Microbiology* 2.9 (Sept. 2017), p. 17121 (cit. on p. 19).

[ZHH08] Achim Zeileis, Torsten Hothorn, and Kurt Hornik. “Model-Based Recursive Partitioning”. In: *Journal of Computational and Graphical Statistics* 17.2 (2008), pp. 492–514 (cit. on p. 76).

[Zel+14] Georg Zeller, Julien Tap, Anita Y Voigt, et al. “Potential of fecal microbiota for early-stage detection of colorectal cancer”. In: *Molecular Systems Biology* 10.11 (Nov. 2014), p. 766 (cit. on p. 25).

[ZLX20] Shuo Zhang, Yang Liu, and Lei Xie. “Molecular Mechanics-Driven Graph Neural Network with Multiplex Graph for Molecular Structures”. In: *arXiv:2011.07457 [physics, q-bio]* (Nov. 15, 2020). arXiv: 2011.07457 (cit. on p. 16).

[Zha+18] Zhongheng Zhang, Heidi Seibold, Mario V Vettore, Woo-Jung Song, and Vieille François. “Subgroup identification in clinical trials: an overview of available methods and their implementations with R”. In: *Annals of Translational Medicine* 6.7 (2018), pp. 122–122 (cit. on pp. 72, 76, 77).

[Zha10] Liping Zhao. “The tale of our other genome”. In: *Nature* 465.7300 (June 2010), pp. 879–880 (cit. on p. 2).

[Zho+19] F. L. Zhou, H. Watada, Y. Tajima, et al. “Identification of subgroups of patients with type 2 diabetes with differences in renal function preservation, comparing patients receiving sodium-glucose co-transporter-2 inhibitors with those receiving dipeptidyl peptidase-4 inhibitors, using a supervised machine-learning algorithm (PROFILE study): A retrospective analysis of a Japanese commercial medical database”. In: *Diabetes Obes Metab* 21.8 (2019), pp. 1925–1934 (cit. on p. 79).

[Zho+18] F. L. Zhou, H. Watada, Y. Tajima, et al. “PDB16 - Compare renal functional preservation outcome of SGLT2 inhibitor in patients with type 2 diabetes: a retrospective cohort study of japanese commercial database with advanced analytics approach”. In: *Value in Health* 21 (2018), S121 (cit. on p. 79).

[ZT15] Jian Zhou and Olga G Troyanskaya. “Predicting effects of noncoding variants with deep learning–based sequence model”. In: *Nature Methods* 12.10 (Oct. 2015), pp. 931–934 (cit. on p. 39).

[Zie+19] Andrzej Zielezinski, Hani Z. Girgis, Guillaume Bernard, et al. “Benchmarking of alignment-free sequence comparison methods”. In: *Genome Biology* 20.1 (Dec. 2019), p. 144 (cit. on p. 9).

Figures list

Tables list

Appendix

A

A.1 Multiple instance learning

Zaheer et al. [Zah+17] defined a function invariant to permutation and place it in a neural network named *DeepSets*. The model is separated into 3 steps. A first neural network Phi projects each instance of the bags into a lower representation. A MIL layer (aggregation operation) invariant to permutation aggregates all instance in a bag. Finally, a last neural network operates on a single instance (aggregated by the preceding step) to compute prediction.

Mathematically Zaheer et al. [Zah+17] defined a property and proved a theorem:

- Invariance to permutation can be formulate like this:

$$f(x_1, ..., x_m) = f(x_{\sigma(1)}, ..., x_{\sigma(m)})$$

  With m the number of elements in the bag and σ any permutation.

- Theorem: A function $S(X)$ operating on a set X can be a valid scoring function i.e it is permutation invariant to the elements in X, if and only if it can be decomposed in the forme $\rho(\sum_{x \in X} \phi(x))$

The last theorem gives a structure for the neural network: ϕ the first neural network, $\sum_{x \in X}$ the aggregation function and ρ the last neural network for classification. The sum operation is trivially invariant to permutation, but we can also define other function like mean or max pooling. Those aggregation functions are quite basic since they are not learned by the network. That's why Ilse et al. [ITW18] have recently proposed a new method to parameterize all transformations. They defined a new function based on an attention mechanism [LPM15]. The proposed MIL pooling is:

$$z = \sum_{k=1}^{|H|} a_k h_k$$

$$a_k = softmax(w^T tanh(V h_k^T))$$

Where $w \in \mathbb{R}^n$ and $V \in \mathbb{R}^{n \times m}$ are parameters and $H = h_1, ..., h_k$ is a bag of k genomes embeddings. To be invariant to permutation the weights' sum must be equal to one.

A.1.1 Beam search strategy using decision tree versus exhaustive algorithm

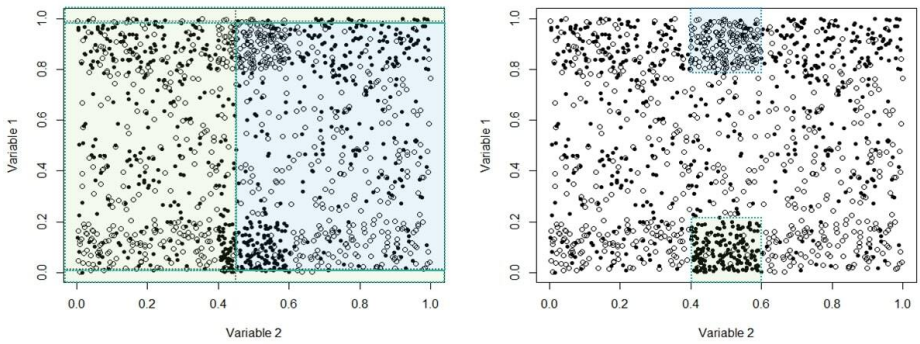


Fig. A.1.: beam search strategy using decision tree versus SD algorithm

A well-chosen example with a categorical target variable (700 empty circles = "No" and 700 filled circles = "Yes") and two numeric description variables. On the left-side, colored areas show decision surface of a three level deep decision tree (green areas related to the "Yes" target, blue area to the "No" target). 4 subgroups are identified:

- Variable 1 $<$ 0.015 (1st split): 76% of Yes, representing 3% of the population
- Variable 1 $\geq$ 0.99 (2nd split): 69% of Yes, representing 2% of the population
- Variable 1 $\geq$ 0.015 and $<$ 0.99 & Variable 2 $<$ 0.45 (3rd split): 52% of Yes, representing 41% of the population
- Variable 1 $\geq$ 0.015 and $<$ 0.99 & Variable 2 $\geq$ 0.45 (3rd split) $\geq$ 54% of No, representing 54% of the population

The 2 last subgroups are of low accuracy in comparison to the target distribution (50%/50% of Yes/No), while the 2 firsts are of low density (less than or equal to 3%). The decision tree did not manage in finding the two subgroups we can easily see with our bare eyes on the right-side (one in the lower center and one in the upper center of the data space), defined as:

- Variable 1 $\geq$ 0.8 & variable 2 $\geq$ 0.4 and $\leq$ 0.6: 91% of No, representing 13% of the population (164 “No” versus 17 "Yes")
- Variable 1 $\leq$ 0.2 & variable 2 $\geq$ 0.4 and $\leq$ 0.6: 92% of Yes, representing 13% of the population (14 "No" versus 166 "Yes").

Both subgroups have higher accuracies than any subgroup from the decision tree. Driven both by a recursive partitioning process and by the interest of overall performance, the decision tree did not capture these regions. For more details, Mario Boley[1] further explores this topic.

A.2 Comparision of *Q-Classifier* with or without cascaded combination of state-of-the-art classifiers

The following box plots aim to analyze, on all studied datasets, the results of the *Q-Classifier* and state-of-the-art classifiers approaches separately and combined in a cascading way. The mean rejection rate is plotted and represents the average rejection rate of all 10 cross-validations. The scores of the *Q-Classifier* come from the accepted samples while the scores of the SOTA classifiers come from the rejected samples and those of the cascade combination concern all samples.

[1] http://www.realkd.org/subgroup-discovery/the-power-of-saying-i-dont-know-an-introduction-to-subgroup-discovery-and-local-modeling/

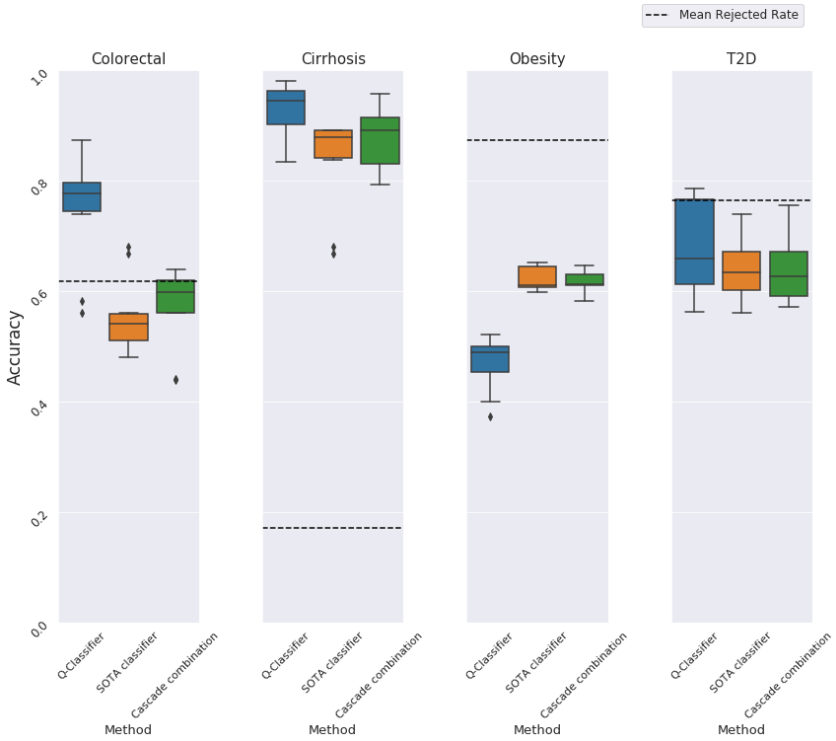


Fig. A.2.: *Q-Classifier* MetaPhiAn2

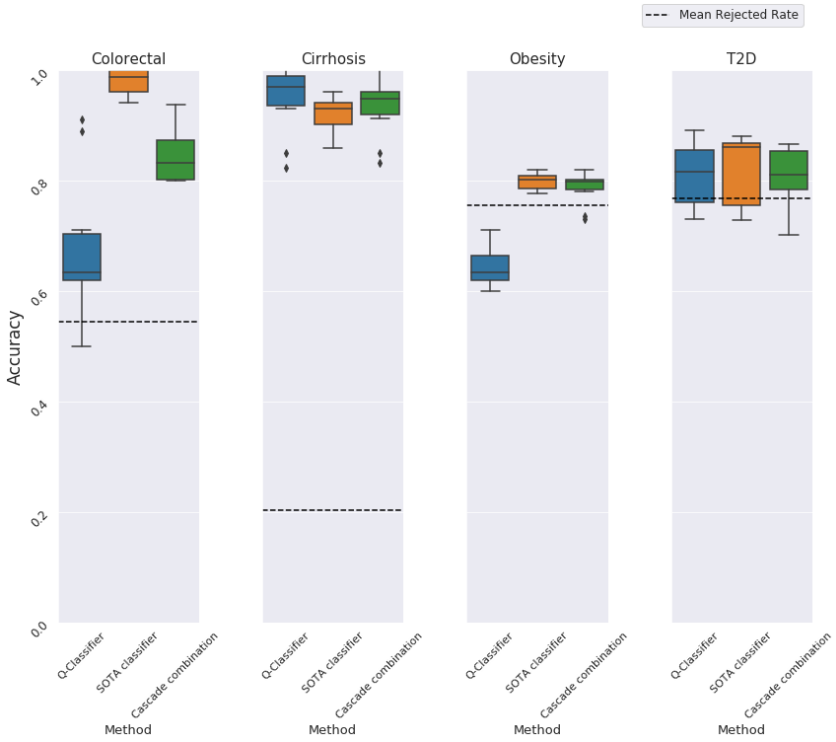


Fig. A.3.: *Q-Classifier* MetaPhiAn2 CLR

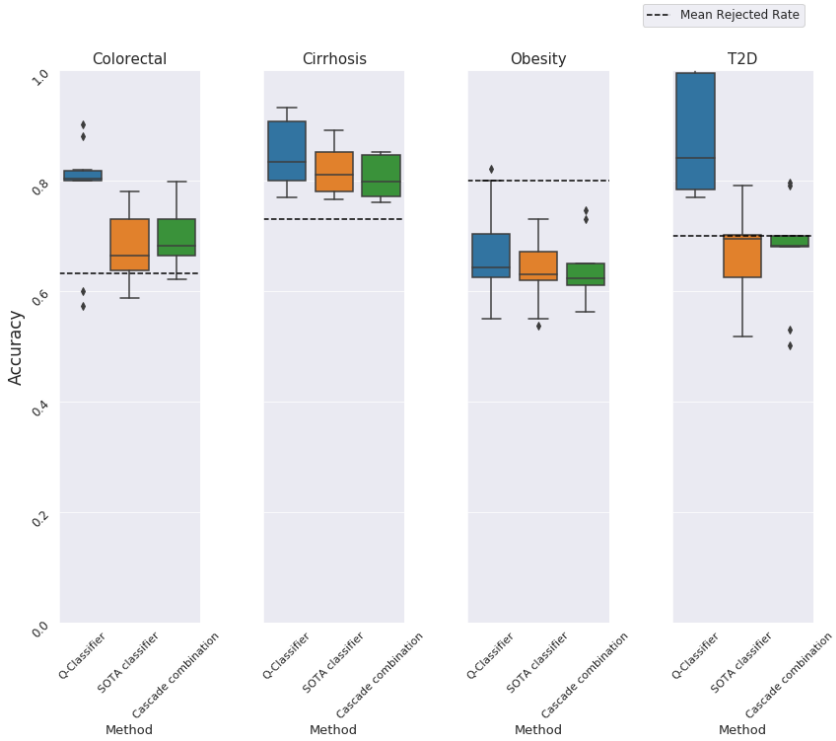


Fig. A.4.: *Q-Classifier* FastDNA

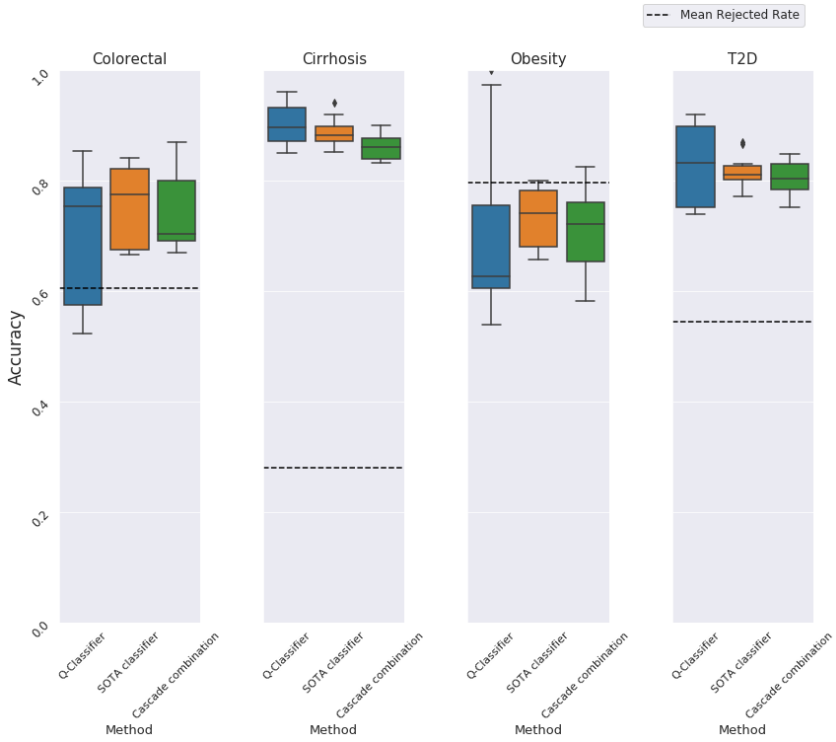


Fig. A.5.: *Q-Classifier* FastDNA CLR

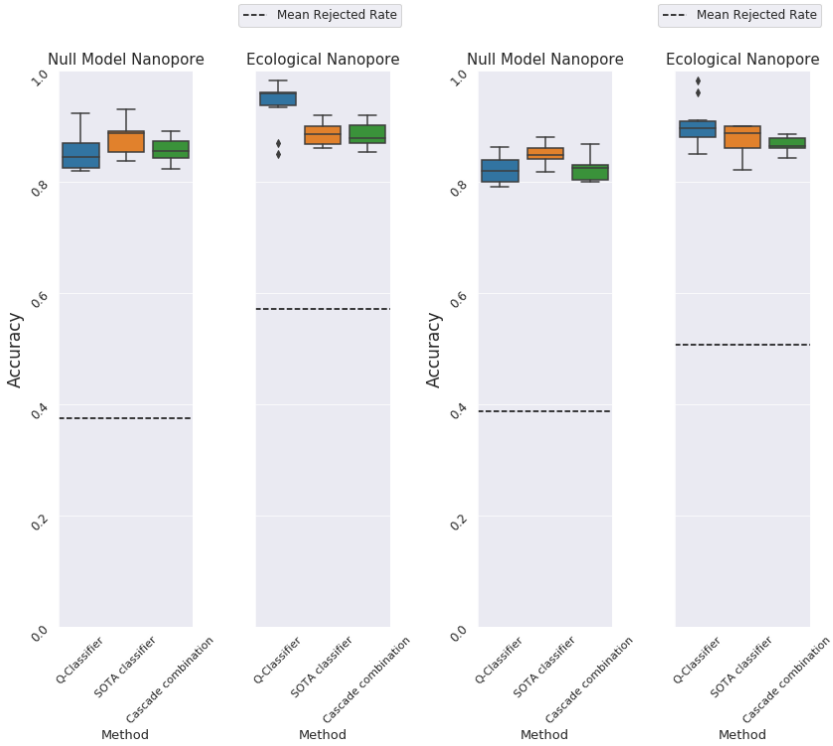


Fig. A.6.: *Q-Classifier* initial abundance

Fig. A.7.: *Q-Classifier* initial abundance CLR

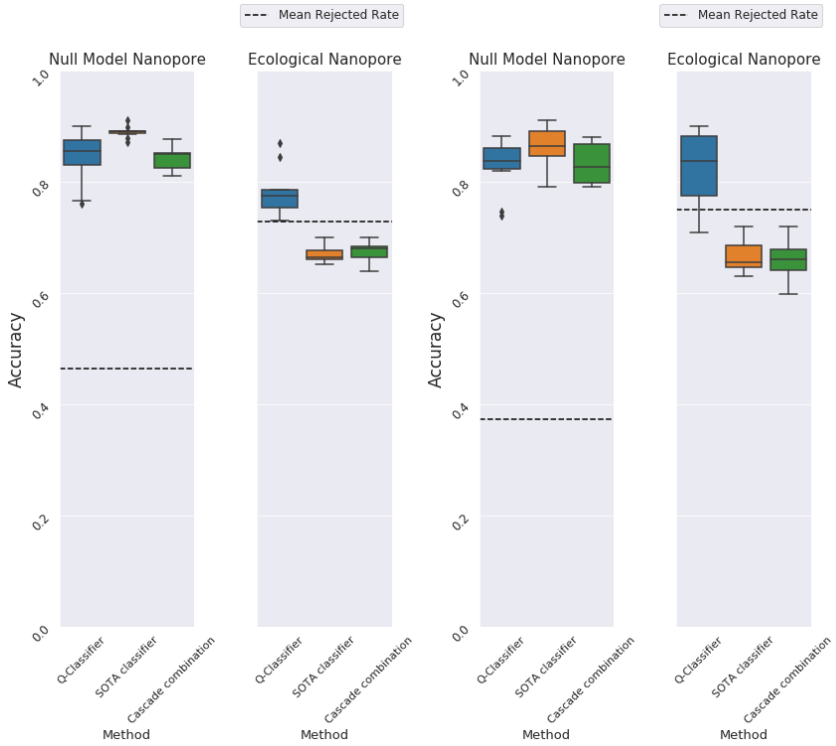


Fig. A.8.: *Q-Classifier* FastDNA

Fig. A.9.: *Q-Classifier* FastDNA CLR

We can see that on most of the datasets, the *Q-Classifier*'s scores are equivalent to the SOTA scores for the examples defined by one or more generated rules. The cascaded combination of the model does not reduce the classification results and seem to be a good compromise between the interpretable prediction of *Q-Classifier* and the black box prediction of the SOTA classifiers.

Glossary

Contig Continuous sequences generated by the alignment of overlapping fragment sequences.. 4

Extrinsic Evaluation An evaluation that focuses on the performance of the finale application.. 44

Intrinsic Evaluation A simple and quick evaluation that focuses on a specific sub task to control the learning of the algorithm.. 47

K-mer Are all substrings of length k contained in a sequence.. 7

Pangenome The pangenome describes the full range of genes in a species.. 8

PCR Polymerase chain reaction, a technique to create copies of a specific DNA region.. 5, 143

Read A sequence of a DNA fragment.. 4

Taxon (plural taxa), it is an entity grouping all living organisms having in common certain well-defined characteristics. The term taxon is used in phylogenetic classification to group living beings according to various criteria.. 11

Colophon

This thesis was typeset with LaTeX 2$_\varepsilon$. It uses the *Clean Thesis* style developed by Ricardo Langner. The design of the *Clean Thesis* style is inspired by user guide documents from Apple Inc.

Download the *Clean Thesis* style at http://cleanthesis.der-ric.de/.