UNIVERSITÉ PARIS-EST MARNE-LA-VALLÉE

RAPPORT DE STAGE

Estimation of abundances in microbial communities from metagenomic data

Author:
Simone PIGNOTTI

Supervisor:
Gregory KUCHEROV

Labex Bézout Master's program
Laboratoire d'informatique Gaspard-Monge

September 2, 2018

UNIVERSITÉ PARIS-EST MARNE-LA-VALLÉE

Abstract

Laboratoire d'informatique Gaspard-Monge

Labex Bézout Master's program in theoretical computer science

Estimation of abundances in microbial communities from metagenomic data

by Simone PIGNOTTI

The computational analysis of genetic sequences has opened the doors to new fields of science at the crossroad of several different disciplines. In metagenomics, genomic data collected from the environment are analyzed to recover the biological composition of the sample, especially the one associated to mibrobial life forms like bacteria, archaea, viruses and fungi. While they have been unexplored until recently, the microbial organisms populating every natural environment, including oceans, soil and human habitats and bodies, play an essential role in the biological activities which are carried out in these ecosystems. They filter waste water, clean pollution and create medicines. In the human body, those same communities have recently been found to be associated with complex diseases like inflammatory bowel diseases, obesity and colorectal cancer.

In this work we present computational methods allowing the characterization of these communities, and we introduce a new approach based on the analysis of short subsequences shared between the sample and a database of known genomes, followed by the fit of a linear model built upon the sequence similarities in the database. We show that this method allows to reliably retrieve the set of microbial species present in the sample and to closely estimate their relative abundances. Furthermore, this method has been implemented in the open-source metagenomic classifier ProPhyle.

Acknowledgements

All my gratitude goes to three special people without whom this work would not have been possible.

Gregory Kucherov, my supervisor at LIGM, who has been mentoring, supporting and having faith in me for over two years and thanks to whom I have made my most significant research experiences. I cannot express how important it was to meet him and collaborate with him for my growth.

Michael Baym, my supervisor during my visit at the Department of Biomedical Informatics of Harvard Medical School, thanks to whom I had the chance to participate in this vibrant research community and learn from him and his wonderful team. His support and advice have been invaluable.

Karel Břinda, friend and colleague, who has been a model for my research career, a patient teacher and collaborator, and who has genuinely helped me to make progress in this path since the very beginning. I am not the only person believing he is a higher form of human being.

I would also like to thank my parents Arianna and Angelo who have always supported every choice of mine, my girlfriend Bérénice for her love and understanding, and all the stimulating people I met during this Master at UPEM and my stay at Harvard Medical School, in particular Siân Owen, Scott Chimileski and Luca Freschi.

Finally, I thank the Labex Bézout program for the scholarship which made this experience possible.

Contents

Chapter 1

Introduction

1.1 Context and Motivations

Even before understanding the structure and function of DNA, humans have advanced hypothesis on how certain traits can be inherited over generations. While there is still much to be learnt about this molecule, the advent of molecular biology and bioinformatics marked the beginning of a new era, deeply changing people's perception of life. Technological advances enabled the analysis of DNA and other biological molecules, whose sequences can now be observed and for which we found a convenient representation as strings over a small fixed alphabet. Such representation enables their computational analysis, often using algorithms which were originally conceived for the analysis of text.

Starting from simple microbial organisms, researchers started to assemble the short genomic sequences produced by sequencing technologies into genes and chromosomes, till the entire human genome has been characterized in 2001. Nowadays, genomic databases like NCBI's GenBank contain billions of sequences, and in the case of GenBank its size is doubling approximately every 18 months [1]. Several algorithmic challenges were introduced by the amount of sequencing data being generated, spanning from compression and storage to annotation and comparison.

The incredible amount of information stored in these sequences is far from being extracted. Algorithmic challenges and technological limitations make the comprehensive analysis of all the produced sequencing data impossible at the current state, and including every such sequence in a genomic "search engine" is unfeasible even with the latest methods and the advances in cloud computing. The potential of these data for understanding human diseases and environmental issues is astonishing, and there is a real need for the development of efficient algorithms and data structures for this most relevant field of Big Data.

Nevertheless, progress in bioinformatics created the possibility to study entire populations and communities of hundreds of different microbial species without the need to isolate single clonal populations or cells. In metagenomics, the genetic material extracted from environmental samples is sequenced and the resulting short fragments of DNA, called *reads*, are assigned to an extensive database of reference genomes. The size of the datasets, the mutations occurring at a fast pace and the presence of organisms which have never been sequenced yet make the use of standard mapping algorithms not fit for this scenario. Therefore new methods are being developed which carry out the task of assigning short sequences to the genomes of the organisms they are most likely to be originated from, using heuristics for speeding up the computation, decreasing the space complexity and estimating the composition of the sample.

[1] https://www.ncbi.nlm.nih.gov/genbank/statistics/

Here we focus on the estimation of relative abundances of microbial organisms in metagenomic samples, and we introduce a method to probabilistically distribute the assignments of the metagenomic classifiers, in particular those of our software ProPhyle. We use a linear regression model to account for the similarities of the reference genomes and solve the problem of reads which map equally well to multiple reference genomes. We show that the method provides accurate results and is comparable to other state-of-the-art abundance estimation tools.

In the rest of this chapter we introduce notions about the analysis of genomic sequences and we provide a background for readers who are not keen to bioinformatics. In Chapter 2 we analyze relevant programs and algorithms performing both read assignment and abundance estimation. In Chapter 3 we introduce the method we propose and finally in Chapter 4 we show the results obtained on different datasets.

1.2 Analysis of DNA sequences

Even though they may sound like fairly simple organisms, the genomes of most bacterial species are composed of millions of nucleotides. Obtaining a faithful representation of them is until today a big challenge, mostly because modern sequencing technologies only provide short fragments (*reads*) of this long chain, usually in the order of few hundred base pairs, or letters; while cutting-edge tools may produce longer sequences, this is at the cost of introducing a considerable amount of sequencing errors. In both cases, reads need to be assembled like puzzles to recover the original genome of the organism, a task which requires a good coverage of the genome and that is made more complex by the presence of highly repetitive portions called *low-complexity regions*. In reference-based metagenomics, a database of high quality reference genomes is needed for the assignments not to be biased by i.e. contamination from other organisms or poor assembly.

Indeed, the reads of the metagenome need to be compared to a set of reference genomes in order to estimate the composition of the sample. The first algorithms to perform the comparison of two sequences were based on dynamic programming: the input sequences were aligned in such a way to minimize the number of mismatches, insertions or deletions to transform one sequence into the other. These algorithms soon became computationally unfeasible due to the fact that unforeseen amounts of data were being generated, that they only allowed pairwise comparison, and that their cost was quadratic in the size of the input sequences. In the current setting, the analysis of metagenomic reads with such a tool would require years; furthermore, the actual alignment is not needed to assess the composition of the sample, and we are mainly interested in which genome the read is originated from.

Soon heuristic methods were developed for sequence comparison, whose BLAST is probably the best representative and is therefore analyzed in Chapter 2. Most importantly, the alignments are performed only for those reference genomes which have an exact match for a subsequence in the read, drastically reducing the number of pairwise comparisons. In the context of metagenomics though, even tools like BLAST, which are still used in other omics fields, became unfeasible for the current size of experiments. Pushing the idea behind BLAST's so-called "*seed-and-extend*" paradigm, alignment-free algorithms started to take its place. In alignment-free sequence comparison, sequences are viewed as sets of substrings of fixed size k, called k-mers, which can be indexed and queried promptly. Reads are then assigned to genomes sharing enough k-mers with them. This similarity measure, surely weaker

than alignment score, allows the analysis of huge metagenomic samples in a fraction of the time: as an example, the software Kraken (also analyzed in Chapter 2) reaches assignment speeds of up to a million reads per minute on databases containing thousands of reference genomes.

This performance is achieved also thanks to the organization of sequences in an evolutionary tree reflecting sequence similarity, called *phylogenetic* tree, which is commonly used in metagenomics to compress the references and provide abundance estimates at different level of the tree. A special kind of phylogenetic tree, called *taxonomic* tree, is often used as it contains annotations like names for different levels of the tree, reflecting the characteristics of the subtree (i.e. species, genera, families). Since genomes in the same species share most of their genetic material, reads may now be assigned to internal nodes of the tree; furthermore, biological mechanisms like horizontal gene transfer, allowing bacteria of different species to exchange portions of their DNA, make assignments even more complex since a short read fragment may match equally well reference genomes associated to distant leaves of the tree.

Estimating abundances solely from the unique mappings to the references has been shown to introduce considerable biases [1] due to the uneven representation of microbial clades in the genetic databases and their variable average similarity within those clades. Therefore methods have been developed to probabilistically redistribute assignments to multiple reference genomes or to internal nodes of the taxonomic tree to fixed ranks, like species or genus. While these methods approximate very closely the abundances of the samples at such ranks, they once again introduce biases linked to the structure of the tree, which is constant subject of discussion and dissent in the microbiology community. For this reason, and to provide better resolution to the results, modern tools focus mainly on the genome level.

Chapter 2

State of the Art

In this chapter we present some of the tools which had a big impact in the field of bioinformatics, and especially metagenomics. Even though we focus on abundance estimation techniques, we provide an overview of the most important methods for the assignment of reads, both based on alignments and on *k*-mer composition, since their properties highly influenced the solutions adopted for the estimation of abundances. This list is not nearly extensive, and we refer to [2] for a more detailed overview of advances in bioinformatics, and to [3] for an extensive list of tools.

2.1 Read Mapping

2.1.1 Alignment-based Methods

- **BLAST** [4]: developed in 1990, this is the first alignment method to be used in practice for metagenomics, namely in the notorious framework MEGAN [5]. Compared to the previous methods solely based on dynamic programming, whose output is the optimal alignment of two sequences (according to some parameters and to the definition of optimality, based on matches, mismatches, insertions and deletions in the alignment), this tool uses heuristics to achieve orders of magnitude higher alignment speed. The way it works is according to the *seed-and-extend* paradigm: first, a subsequence of the read to align is queried in an index containing informative subsequences of the reference genomes (*seeding* step), then if an exact match is found the program extends it using an alignment algorithm (*extending* step). If the alignment score drops below a certain threshold, the alignment is interrupted and another seed is used. This last optimization has a huge drawback when the reads come from a metagenome: in the presence of a novel organism which is not included in the reference database, BLAST will stop the query reporting no alignment, decreasing the overall sensitivity of the method. A desirable property of aligners for metagenomics would be to report even low-quality alignments with genomes similar to the query, which can still provide useful information on the composition (especially if the references are organized in a taxonomic tree). Even though the threshold can theoretically be reduced, the trade-off between sensitivity and time complexity often leads to unsatisfactory results. Furthermore, the use of an extensive reference database requires TBs of memory to store the huge hash table used to store the seeds. Even though several other alignment algorithms have been developed during the 90's and the beginning of this century, BLAST has been keeping to attract users till nowadays: it and its numerous optimizations are still used extensively in various scenarios, including metagenomics.

- **BWA-MEM** [6]: one of the most used alternatives to BLAST is currently BWA framework's aligner BWA-MEM. The algorithm is based on identifying maximum exact matches (MEM) between the suffixes of the read and the reference sequences. Those matches are used as seeds and extended similarly to what is done in BLAST. The main innovation brought by this tool is the use of an efficient indexing structure, the BWT-index. This data structure, first introduced for full-text search applications, makes use of a reversible text permutation, the Burrows-Wheeler transform, which reduces the entropy of the sequence to be indexed by grouping runs of the same characters. While this is especially useful for compression, it also has a deep relationship with suffix arrays and, thanks to an additional data structure, it allows for queries of a pattern of length p occurring occ times in the reference in time $\mathcal{O}(p + occ)$. This efficient index has become a mainstream data structure in bioinformatics, and several other tools are based on it. In addition to being very memory efficient, BWA is also much faster than BLAST: aligning 1M reads to the first 9 chromosomes of the human genome, for instance, takes BWA only 6 minutes, while BLAST requires more than 24 hours.

2.1.2 Alignment-free Methods

- **Kraken** [7]: this is among the first alignment-free tools developed for metagenomic classification, and it set a new standard for assignment speed, allowing to analyze up to one million metagenomic reads per CPU minute. In addition to the set of reference sequences, Kraken uses the taxonomic tree on which they are clustered to compress the database and speed up the queries. Its index consists of a hash table associating each k-mer to the lowest common ancestor (LCA) of the reference genomes containing it in the taxonomic tree. This data structure provides very fast queries, but requires large amounts of memory. Furthermore, the LCA statistics requires a fundamental assumption to be verified by the tree topology: a k-mer associated to an internal node should be present in most reference genomes of its subtree. This is far from being true, because of phenomena like horizontal gene transfer which allows bacteria of different species to "exchange" portions of their genomes, and because taxonomic trees are built not only upon actual sequence similarity, but also on morphology, history and geography of their discovery, harmness to humans etc. As a consequence, propagating a k-mer present in only two genomes which are very distant in the tree to its LCA introduces a considerable number of false positives, since in the database this k-mer will be considered as present in every genome in the subtree of the LCA. In conclusion, even though Kraken has opened new possibilities in metagenomics thanks to its classification speed, its applications are now limited because of the amount of memory it requires and the LCA heuristics getting less and less accurate as more genomes are sequenced and included in the reference databases. As explored in [8], at the current state the NCBI RefSeq database requires 2TB of memory and 11 days to be indexed by Kraken, and while the percentage of classified sequences increases considerably with the expansion of the database, the overall accuracy of the method decreases, most likely because of flaws in the LCA assumption when tens of thousands of reference genomes are indexed.

- **ProPhyle** [9]: addressing the main issues in Kraken and its legacy, ProPhyle moves forward from the LCA heuristics to a lossless index where each k-mer

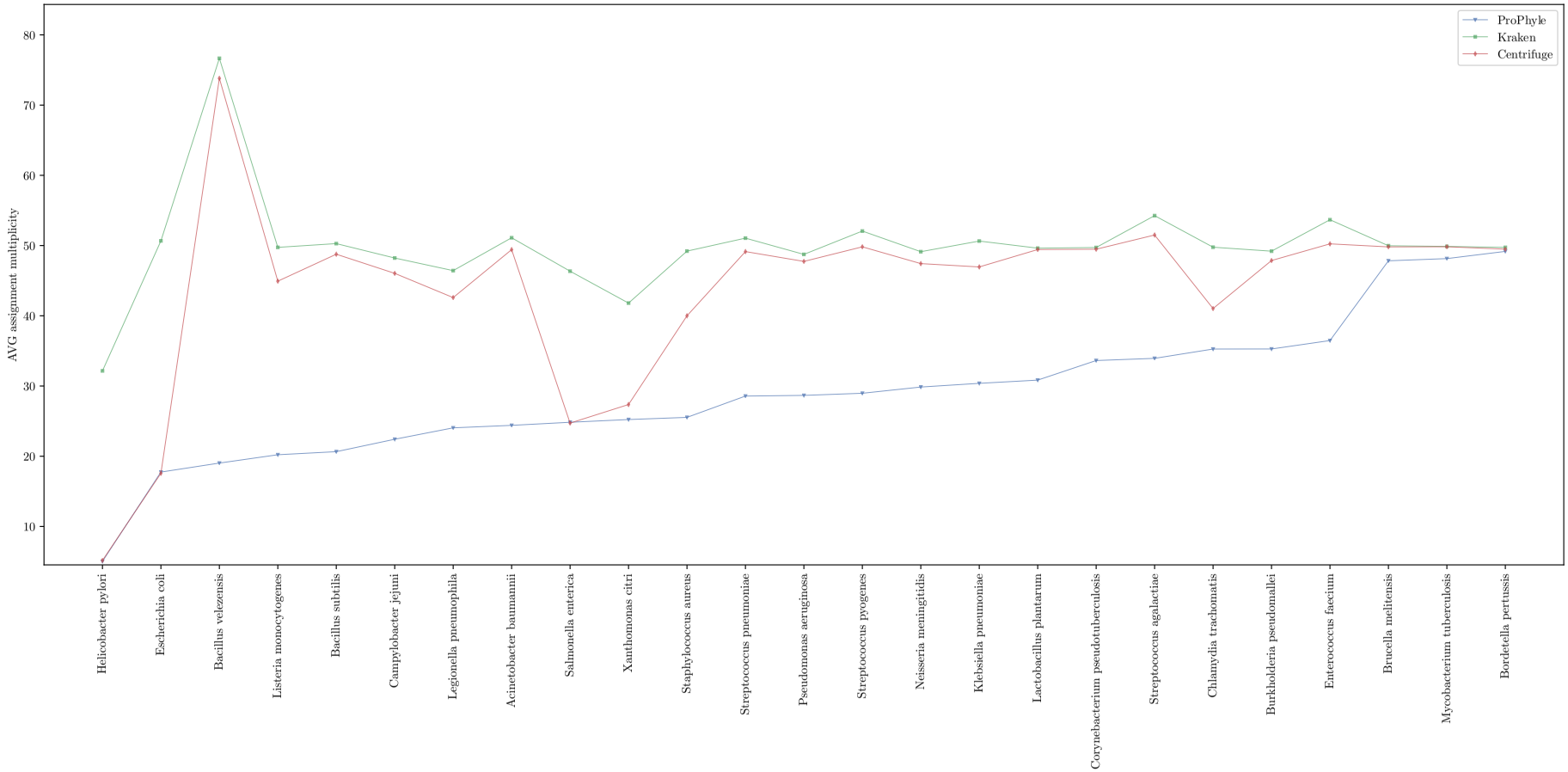
is associated to the exact set of genomes where it appears. In doing so, it surprisingly reduces the amount of memory required both for the creation of the index and the classification of sequences with respect to Kraken. The same reference database, containing around 2700 bacterial genomes, which would require 90GB of memory for the construction of a Kraken database and 70GB for the classification of reads, will only require under 16GB with ProPhyle, fitting a laptop's resources. This program propagates *k*-mers up in the taxonomic tree only if they are shared by every child of a given node, and compresses the set of *k*-mers associated with a node using an assembly algorithm based on the greedy enumeration of disjoint paths in a De-Bruijn graph. While this process enables the compression of the *k*-mer information required from the reference sequences, the topology of the tree only plays a role in defining the compression rate and the speed of the classifier, but not on its accuracy; furthermore, the choice is not limited to commonly used taxonomic trees as the NCBI one used by Kraken, but any phylogenetic tree in the Newick format can be used. The collection of sequences resulting from this compression process is indexed using the BTW-index, providing slightly slower queries than Kraken but considerably reducing the memory requirements while providing more accurate assignments as shown in figure 2.1. In particular, reads matching several nodes with the same score are assigned to the entire set, and the output contains the exact information about each *k*-mer match and mismatch for every node.

2.2 Abundance Estimation

2.2.1 Bayesian Re-estimation

- **Bracken** [1]: the purpose of this tool, whose name stands for Bayesian Reestimation of Abundance after Classification with KrakEN, is to reduce the bias due to the lossy database of Kraken by estimating the probability that reads from a reference genome are assigned to its ancestors in the taxonomic tree. Using Bayes' theorem, this information is inversed to estimate the probability that a read assigned to an internal node comes from the leaves (genomes) in its subtree, enabling the re-distribution of those reads to the genomes. In the publication, the authors show that the tool is able to achieve a satisfying estimation of abundances of two notoriously complex species, Mycobacterium Bovis and Mycobacterium Tuberculosis, whose genomes are characterized by >99.5% similarity. Nevertheless, this approach has a main drawback: it introduces a vast number of false positives by redistributing reads assigned to internal nodes to genomes which were not in the sample in the first place; even though users can specify a threshold for the minimum number of specific assignments that a genome needs to have to be considered as present in the sample, this does not allow to correctly filter false positives because of the combination of the LCA heuristic and biases in the representation of different species, where some have thousands of representative genomes and therefore it is less likely to have specific assignment to a single one, while others have only one representative.

FIGURE 2.1: Average assignment sets cardinalities for ProPhyle, Kraken and Centrifuge. The index for the tree tools were built from a set of 50 genomes from each of the 25 species listed in the x-axis, for a total of 1250 genomes. Reads were simulated from each genome and assigned to the index. On the y-axis is the average size of a read's assignment set (number of genomes in the subtrees when assignments are to internal nodes). ProPhyle's assignments are more specific than those of the other tools, making it a better candidate for metagenomic classification with indexes including several thousands of similar reference genomes.



2.2.2 Expectation-Maximization

- **Kallisto** [10]: born as a RNA-seq quantification program, Kallisto has been applied to metagenomics in its variant Metakallisto. After assigning reads to equivalence classes, i.e. set of reference genomes likely to have generated them computed by intersecting the k-equivalent classes of each k-mer they are composed of, Kallisto runs an Expectation Maximization (EM) algorithm, inherited by its predecessors Cufflinks and Sailfish, to estimate the real abundances from the counts of assignments to equivalence classes. The EM algorithm iteratively optimizes the likelihood of parameters in a statistical model. It alternates between two steps: in the E-step it calculates the likelihood with the current parameters, and in the M-step it sets the parameters to maximize its value. An example of definition of such model is given for Centrifuge.

- **Centrifuge** [11]: the EM algorithm applied to RNA-seq quantification influenced several other metagenomic classifiers, including MetaMaps [12] and Centrifuge. This tool is based on the BWT-index and on probabilistic propagation of subsequences of the reference genomes in a taxonomic tree, and computes pseudoalignments with an heuristic seed-and-extend algorithm. Starting from these pseudoalignments, the likelihood of an abundance vector at i.e. the species level is defined as follows:

$$L(\alpha|C) = \prod_{i=1}^{R} \sum_{j=1}^{S} \frac{\alpha_j l_j}{\sum_{k=1}^{S} \alpha_k l_k} C_{ij}$$

where R is the number of reads, S is the number of species, α_j is the abundance of species j, l_j is the average length of the genomes of species j, and C_{ij} is 1 if read i is classified to species j and 0 otherwise. The two steps of the EM algorithm are as follows:

E-step: estimate the number of reads n_j assigned to species j with the current abundance configuration:

$$n_j = \sum_{i=1}^{R} \frac{\alpha_j C_{ij}}{\sum_{k=1}^{S} \alpha_k C_{ik}}$$

M-step: update the estimated abundance of species j (α'_j), which will be used as α in the next iteration:

$$\alpha'_j = \frac{n_j / l_j}{\sum_{k=1}^{S} n_k / l_k}$$

This approach overcomes some of the problems of Bracken and previous methods for abundance estimation, but the non-parametric definition of the likelihood function does not allow to push the model towards fitting metagenomic samples of different complexities. Furthermore, even though the EM algorithm is known for its fast convergence, in this situation the optimization of the likelihood may require more time than the classification step, since the number of dimensions is equal to the number of reference genomes in the index and the optimization only stops when $\sum_{j=1}^{S} |\alpha_j - \alpha'_j| < 10^{-10}$.

2.2.3 Linear Models

- **GASiC** [13]: Genome Abundance Similarity Correction implements a simulation-based approach to estimate the similarity of the reference genomes and defines a regularized linear model for the correction of read alignments. Compared to other tools like Megan [5], which employ the structure of a phylogenetic tree to resolve ambiguous read mappings of alignment algorithms or ignore multiple assignment thus introducing severe biases, GASiC's model is only founded on the similarity of the references, as perceived trough the specific read sequencing technology and aligner used. Our work is mainly inspired by this publication, and overcomes the issues linked to scalability and mapping sensitivity due to the alignment step. The description of the linear model is part of the methods section.

- **DiTASiC** [14]: standing for Differential Taxa Abundance including Similarity Correction, this program extends and improves GASiC's model (from the same research group) and introduces a statistical model for differential abundance analysis, e.g. how reference genomes are differently expressed in samples collected over time or in different locations. While the linear model used for the correction of mappings is inherited from GASiC, DiTASiC optimizes it using a Poisson Generalized Linear Model (GLM) with identity link function, which is more suitable for count data. On the other hand, our tests in Chapter 4 show that the tool overestimates the number of reference genomes in the samples, either because of the inaccurate assignments provided by the Kallisto pseudo-alignment framework or for the lack of regularization in its model. Indeed, the authors recommend to use other tools to estimate the composition of the sample prior to constructing DiTASiC's index, to only include the references which are most probably present and reduce the amount of false positives.

Chapter 3

Methods

3.1 Context

To cope with the complexity of metagenomes, the size of the reference databases and the resulting uncertainty in read assignments, early abundance estimation tools like Bracken [1] and MEGAN [5] did not provide the abundances for each reference genome, but only for taxonomic clusters like species, genus or phylum.

This approach introduces a bias due to the structure of the phylogenetic or taxonomic tree, in which some species may be over-represented, others are still subject to adjustments, and in certain cases the definition is not coherent: in the Mycobacterium genus, for instance, Mycobacterium Bovis and Mycobacterium Tuberculosis are classified as two different species even though their genomes are characterized by 99.5% similarity, while the threshold used for species separation is normally way lower [15].

Furthermore, the technological improvements in read sequencing provide increased resolution for the metagenomic samples, by generating either short high quality reads with error rate <1% as with Illumina and Sanger machines, or long reads with higher error rates as with ONT and PacBio. The detail of the information in these sequences allow, with sufficiently high coverage, to map at least some of the reads in the metagenome to the exact reference genome they are likely to come from, when they are present in the database.

Nevertheless, due to the high similarity among the reference genomes of the same species, most reads, especially those associated to the core genome, will still map to multiple references equally well. Therefore reads assigned to subsets of the reference database need to be probabilistically redistributed to each genome.

Several tools including Bracken and GASiC build their model for read redistribution on top of the estimated similarity of the reference genomes. In Bracken, every k-mer of each reference is mapped to the index using the Kraken algorithm, and the frequency with which k-mers are assigned to internal nodes of the taxonomic tree are used to estimate the probability that a read of the metagenome assigned to such internal nodes originates from each reference in its subtree using Bayes' theorem. In this model, the k-mer length k should match the length of the reads in the sample, but the technology used to sequence the metagenome is not taken into account, reducing the robustness of the method. By fixing k, it also makes it unfit for machines producing reads of variable length like ONT. Furthermore, by assigning every k-mer of the database it increases the computational burden and the risk of overfitting.

Gasic [13] introduced a new approach based on read simulation. Read simulators provide realistic sequences by using probabilistic models built from the results of different sequencing technology. This allows the abundance estimation to be independent from the sequencing technology, which evolves at a fast pace. Reads

are simulated from each reference genome with a low coverage and with parameters resembling those of the metagenomic sample, then aligned to the references using a fast aligner like BWA [6] or Bowtie [16]. Reads mapped to multiple reference genomes are used to build a similarity matrix which is specific to the sequencing technology and the aligner used in the real experiment, and a LASSO linear model is built on top of it.

Inspired by these two methods, we developed an abundance estimator for alignment-free tools, which has been integrated in the program ProPhyle. Exploiting ProPhyle's lossless k-mer index, producing assignments which are more accurate than Kraken's (see figure 2.1), we decided to focus on the genome level and to make use of the model introduced by Gasic for alignment tools.

3.1.1 Desired Properties

A major problem with Bracken and other abundance estimators is the amount of false positives in their output. Indeed, due to the size of modern metagenomic read sets, which may contain up to billions of sequences, it happens frequently that reference genomes which are known not to be present in the sample still have non-zero assignment counts. This happens either because those genomes are highly similar to other references which indeed are present in the sample, or because those reads are generated from a novel strain which is not included in the database and has no better representative, in which case the assignment will probably have a low score.

While dealing with the second issue requires a preprocessing step to filter out poor quality assignments, the first one can be solved by defining thresholds for the number (or percentage) of specific assignments that a given genome needs to get to be considered as present, where specific means that no other reference genome is matched equally well, as it is done in Bracken. Nevertheless, this solution is once again affected by the uneven representation of species in the database, where well-studied species like human pathogens include hundreds or thousands of representative genomes while others only have few representatives. In the first case, the probability that a read matches an exact genome instead of an entire cluster decreases considerably, since most k-mers are shared by the whole cluster.

The solution adopted in Gasic, using LASSO [17] to regularize the linear redistribution model and filter out false positives, also has some drawbacks. The major issue appears in the presence of highly similar clusters in the reference database: if the genomes in the cluster all have a comparable amount of assignment, the L_1 regularization penalty will be likely to select one of them at random and filter out the others.

To address the issue of Gasic, we used the Elastic Net regressor. Elastic Net combines the L_1 regularization used in LASSO and the L_2 regularization (also called Tikhonov or Ridge regularization). The Elastic Net can be seen as a generalization of LASSO, and it introduces a so-called "grouping effect", avoiding the random selection of highly correlated genomes as it happens with LASSO. Furthermore, the path with which predictors are set to 0 by its variable selection property is more stable than LASSO's when regularization parameters change (figure 3.2). This feature guarantees consistent results when using different parameters to fit metagenomes of different complexity.

3.1.2 Linear Regression

Linear regression is a linear approach to modelling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables) [18]. The relationship between these two sets of variables is modeled with linear functions of the kind:

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i \,, \quad i = 1, \ldots, n$$

where y_i is the response variable, the vector $\mathbf{x_i} = \{x_{i1}, \ldots, x_{ip}\}$ contains the p explanatory variables for y_i, the vector $\boldsymbol{\beta} = \{\beta_0, \ldots, \beta_p\}$ is the vector of regression coefficients including the intercept term β_0, and ε_i is an error term which accounts for noise in the data. In matrix notation, the model can be expressed as:

$$\mathbf{y} = \boldsymbol{X\beta} + \boldsymbol{\varepsilon}$$

The standard method for estimating the regression coefficients is Ordinary Least Squares (OLS). This method consists in minimizing the sum of the squares of the differences between the observed dependent variable and those predicted by the linear function [18]. This is equivalent to the optimization of the following function:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\beta} \|\mathbf{y} - \boldsymbol{X\beta}\|_2^2$$

Under the assumption that the error vector $\boldsymbol{\varepsilon}$ is normally distributed, OLS is therefore equivalent to the Maximum Likelihood (ML) estimator [18].

In the context of metagenomic abundances, the observed assignment counts can be modelled as the dependent variables of a linear model, while the regression coefficient could represent the unknown abundances of the reference genomes in the sample, to be estimated using some explanatory matrix. In our case, the explanatory variables will be measures of similarity of the reference genomes. Due to the similarities between genomes of the same species being very high, and to the increasing number of genomes used as references in metagenomic experiments, it is unrealistic to expect this model to fit the data by simply using OLS. In fact, in this case most of the genomes will be considered as present by the model, since the reads in the sample will be mapped to several references, but only few of them will actually be present. To address this issue, we use a model which performs both regularization and variable selection: the Elastic Net.

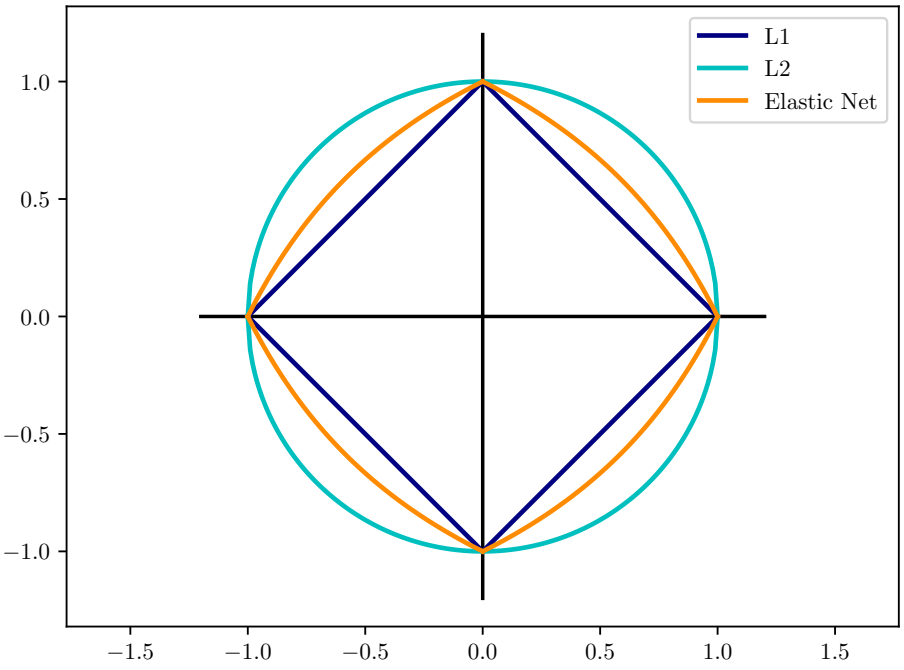
3.1.3 The Elastic Net

To avoid overfitting a linear model, or to improve its performance when the problem is ill-posed, the technique of regularization is often used. In a regularized model, a regularization term is included in the minimization we have already seen for OLS:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\beta} \|\mathbf{y} - \boldsymbol{X\beta}\|_2^2 + \|\boldsymbol{\Gamma\beta}\|_2^2$$

where $\boldsymbol{\Gamma}$, called Tikhonov matrix, is usually chosen as a multiple of the identity matrix. Although introducing this L_2 regularization penalty, so called because it uses the l^2 norm or Euclidean distance, has a beneficial effect on overfitting issues by shrinking large regression coefficients, an L_1 regularization penalty can be used

FIGURE 3.1: Optimization space for the L_1, L_2 and Elastic Net regularizations in 2D. Adapted from [20]



in order to enforce their sparsity, thus performing variable selection. This technique, known as LASSO [17], has been already proved to be an effective solution for metagenomics in [13].

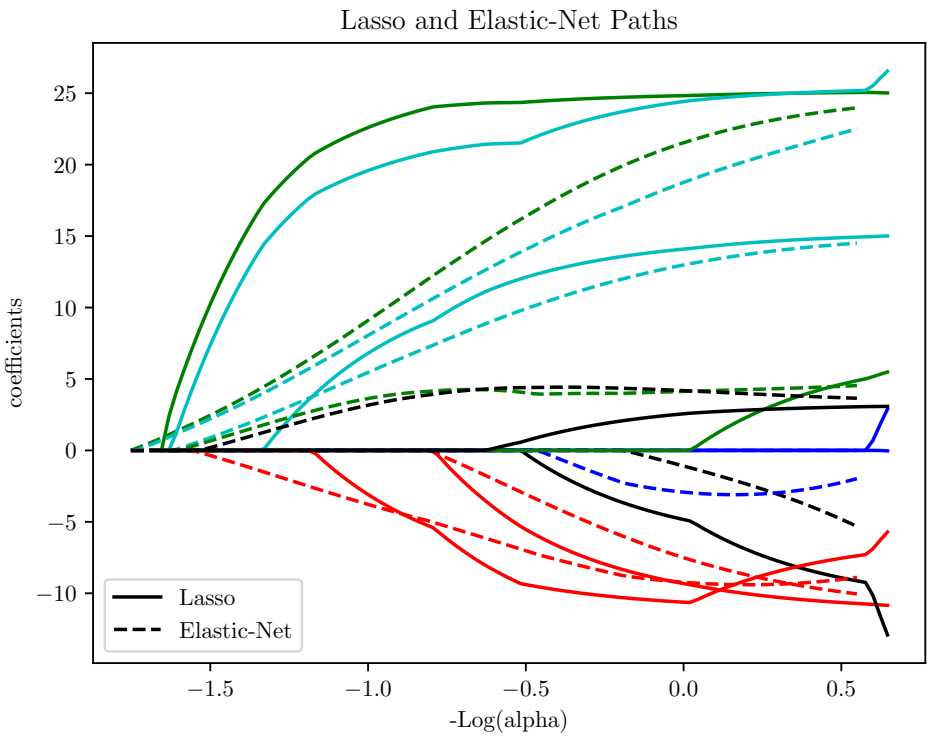
While the most intuitive way to enforce sparsity is by using the l^0 norm, i.e. the number of non-zero coefficients, this would break the convexity of the objective function to optimize, making it an NP-hard problem [19]. Nevertheless, the l^1 norm has been shown to approximate l^0, therefore providing similar performance without loosing the convexity property. On the other hand, the l^2 norm does not provide the same effect: the intuition of the reason in 2D is in the figure 3.1. Let us consider the problem as a constrained optimization problem:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\beta} \|\mathbf{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 \text{ s.t. } f(\beta) \leq t$$

for $f = \|\cdot\|_1$ and $f = \|\cdot\|_2^2$. In the first case, the likelihood of a convex object which lies tangent to the boundary to encounter a "corner" is higher than in the second case, where the rotational invariance of the (n-)sphere contrasts this property.

While both L_1 and L_2 regularization penalties have desirable properties for our application, respectively the sparsity and the shrinking of the regression coefficients, they both come with some drawbacks. The LASSO has proven to be extremely variable because of its inherent discreteness, as shown in [21]. On the other hand, the Tikhonov regression does not provide the desired variable selection property, since using the l^2 norm it keeps all the predictors in the model. It has also been observed that in a variety of applications, each technique may perform better than the other.

FIGURE 3.2: Example of regularization path for the coefficients of the LASSO and Elastic Net methods on the same instance. Varying the regularization weight α results in smoother transitions for the Elastic Net. Adapted from [20]



The Elastic Net addresses the issues above by linearly combining the L_1 and L_2 regularization penalties. The resulting objective function to optimize is:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\beta} \|\mathbf{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda_1\|\boldsymbol{\beta}\|_1 + \lambda_2\|\boldsymbol{\beta}\|_2^2$$

The greatest advantage provided by this method is its grouping effect. As shown in Theorem 1 of [22], this property tends to give similar regression coefficients to highly correlated variables. This is particularly important when fitting metagenomic data of high complexity, where several strains of the same species may be present in the sample. In such situation, LASSO would be likely to only select one such variable at random [22].

3.2 Model

Inspired by the LASSO linear model defined in [13] and the improvements in its successor [14], we have designed a model to estimate metagenomic abundances from read assignments, focusing especially on the properties of the assignments of the metagenomic classifier ProPhyle.

Thanks to the lossless property of ProPhyle's index, its read assignments have a higher resolution than other alignment-free methods (cf. 2). Even though metagenomic reads can still be assigned to internal nodes of the taxonomic tree, this has a different meaning with respect to the assignments of e.g. Kraken: for ProPhyle, a read is assigned to an internal node if and only if every genome in its subtree has the same assignment score (i.e. shares the same number of k-mers with the read); for Kraken, this may happen if two or more, possibly distant genomes have the same score, in which case the tie would be solved by assigning the read to their lowest common ancestor (LCA) instead of providing their list, or if they share every k-mer in the read, in which case the k-mers themselves would be associated to their LCA in the database, making a precise assignment impossible in the first place.

Therefore we designed a model which focuses on the genome level, and which does not take into account the structure of the taxonomic tree but only the actual similarity of the reference genomes, as it is "perceived" through ProPhyle's index and assignment algorithm. As introduced in [13], a first simulation step is performed in order to estimate such similarities, then read assignments are optimized using an Elastic Net-based linear model.

3.2.1 Model Definition

Similarity matrix estimation

In order to estimate the similarity between each couple of reference genomes, reads are simulated from each of them and assigned to the full reference set. The similarity matrix, whose columns encode the distribution of the assignments, is built as follows:

$\mathbf{S} \leftarrow \mathbb{M}_{n \times n}$
for ref. genome $i \in D$ **do**
  $s_{*i} \leftarrow \mathbf{0}^n$
  simulate read set R_i from genome i
  for read $\in R_i$ **do**
    $I \leftarrow$ assign(read) $[I \subseteq D_n]$
    for $j \in I$ **do**
      $s_{ji} \mathrel{+}= 1$
  $s_{*i} \mathrel{/}= s_{ii}$

where D_n is the reference database containing the n reference genomes.

The value of s_{ij} encodes the number of reads simulated from reference j which map to reference i, divided by the number of reads which map back to a subset containing reference j. This represents an estimate for the similarity of genomes (i, j), according to their k-mer composition encoded in the ProPhyle index used for the assignment.

Optimization

Let $\mathbf{m} = [m_1, \ldots, m_n]^T$ be the vector of mappings of a sample, calculated in the very same way as a column of $\boldsymbol{S}$ (its sum will exceed the total number of reads in the sample, since a multiple assignment to subset I will increment by 1 the entry of each reference genome $g_j \in I$). In order to recover the vector of true abundances $\mathbf{r} = [r_1, \ldots, r_n]^T$, we need to solve the following system of linear equations:

$$\mathbf{m} = \boldsymbol{S} \cdot \mathbf{r}$$

with non-negativity constraints $r_i \geq 0 \; \forall i$. The equation corresponding to genome i is:

$$m_i = r_i + \sum_{i \neq j}^{n} s_{ij} \cdot r_j$$

To estimate the solution to this system, we solve it using $\mathbf{r}$ as the regression coefficient vector of the Elastic Net.

3.2.2 Implementation

Simulating reads which reflect the properties of the metagenomic sample to be classified is a crucial step of the optimization pipeline: parameters like read length and error profile of the technology used to sequence the real sample may highly affect the performance of the method. For this reason, we have used the simulation framework RNFsim [23] in the implementation of this step. RNFsim includes some of the most efficient and accurate read simulators and enables reproducible and fully-parametrizable read simulation; it can also easily be extended to match the characteristics of evolving sequencing technologies. In order to obtain accurate results, the simulated reads should match the characteristics of the real data as closely as possible.

For the optimization step, we relied on the implementation of the coordinate descent algorithm for Elastic Net provided by the scikit-learn Python framework. The code[1] which performs the optimization is implemented in C as a Cython extension, and provides fast and stable convergence.

[1] https://github.com/scikit-learn/scikit-learn/blob/master/sklearn/linear_model/cd_fast.pyx

At the time of writing, all the programs and scripts are publicly available in ProPhyle's Github repository[2] in the branch `enet_abundances`. Every step in the pipeline described above will be integrated in the next main release of ProPhyle.

3.2.3 Error estimations

To benchmark the method we used the two following error measures, where e_i and t_i are respectively the estimated and true abundances of genome (or species, genus, etc.) i:

- MAE (Mean Absolute Error):

$$\frac{\sum_{i=1}^{n} |e_i - t_i|}{n}$$

- RSS (Residual Sum of Squares):

$$\sum_{i=1}^{n} (e_i - t_i)^2$$

In addition to these measures of the distance between the true and estimated abundances, we focused on the false positive and false negative rates, which are arguably the most important measures in many metagenomic experiments: while slight changes in the relative abundances may be interesting for some particular applications, having a reliable list of what the sample is composed of is always a valuable information. Furthermore, we provide two measures for those rates: first, the number of FP (resp. FN), i.e. the amount of genomes which are present in the output of the programs but not in the sample (resp. present in the sample and not in the output), and secondly their relative abundance. We will show that our method achieves superior genome selection performances with respect to the other tools analyzed, while improving the error calculated with the measures above.

[2] https://github.com/prophyle/prophyle

Chapter 4

Results and Discussion

4.1 Results

To evaluate the performance of this abundance estimation framework, we used both simulated readsets, in which the reference genome generating each read is known and the exact abundances can be calculated, and a real metagenomic sample generated artificially from even volumes of DNA of different organisms.

4.1.1 Simulated Reads

The following results were obtained running our method (with two different parameter sets), Bracken and DiTASiC on the read set from [24], a realistic simulated metagenome originally created for the assessment of performances of assembly algorithms.

The index used for the assignments includes 1267 genomes from 500 species, including the 193 genomes from 85 species in the read set. The same database was used to build indexes for Kraken and DiTASiC. Kraken's results have been further processed using the program Bracken, which redistributes reads assigned to internal nodes of the taxonomic tree to a fixed taxonomic rank (e.g. species, genus, etc.). The error estimations and false positive/negative rates are provided for every such rank till family, where the results of our model are virtually perfect. The results for Bracken at the genome level are not available since this tool is limited to the analysis at species level or above. Both a combination of L_1 and L_2 regularization penalties and a pure LASSO approach have been tested for our method, each providing results of different biological interest.

The table below contains the errors, calculated with the measures introduced at the end of chapter 3, of the three tools. In the row corresponding to false positive (resp. negative) rate, the first integer represents the number of variables (e.g. genomes, species, genera etc.), while the float between parentheses represents their estimated (resp. real) relative abundance. The figure 4.1 shows the results of the tools compared to the ground truth, represented with black horizontal lines.

Rank	Measure	$\lambda = 1$ $\alpha = 0.01$ (LASSO)	$\lambda = 0.998$ $\alpha = 0.003$ (E. Net)	Bracken	DiTASiC
Genome	MAE	7.31e-4	**6.33e-4**	NA	8.52e-4
	RSS	**7.52e-4**	9.32e-4	NA	2.26e-2
	FN # (ab.)	66 (1.06e-2)	**2** (**2.02e-4**)	NA	32 (2.07e-2)
	FP # (ab.)	**32** (**5.60e-2**)	89 (8.57e-2)	NA	648 (3.22e-1)
Species	MAE	3.20e-4	**8.67e-5**	1.52e-4	1.25e-3
	RSS	3.84e-5	**6.68e-6**	2.79e-4	4.53e-2
	FN # (ab.)	**0** (**0**)	**0** (**0**)	1 (1.63e-2)	0 (0)
	FP # (ab.)	**1** (**1.44e-4**)	26 (1.36e-3)	170 (9.73e-4)	381 (6.08e-2)
Genus	MAE	3.01e-4	**4.98e-5**	2.63e-4	1.97e-3
	RSS	1.33e-5	**1.16e-6**	2.85e-4	5.45e-2
	FN # (ab.)	**0** (**0**)	**0** (**0**)	1 (1.63e-2)	0 (0)
	FP # (ab.)	**0** (**0**)	12 (3.62e-5)	78 (3.23e-4)	194 (1.06e-2)
Family	MAE	3.22e-4	**6.04e-5**	3.85e-4	2.10e-3
	RSS	1.35e-5	**1.29e-6**	2.86e-4	5.47e-2
	FN # (ab.)	**0** (**0**)	**0** (**0**)	1 (1.63e-2)	0 (0)
	FP # (ab.)	**0** (**0**)	7 (6.08e-6)	42 (4.51e-5)	108 (4.73e-3)

Since the reference genomes were only a fraction of those used for real experiments, and the sample dataset was not nearly as complex as a real one, the best variable selection for ranks higher than genome is achieved by using only L_1 regularization (first column). Nevertheless, introducing a very small coefficient for L_2 increases the overall precision and reduces the number of false negatives at genome level, probably thanks to the "*grouping effect*" of the Elastic Net. This may be way more beneficial on real instances, where the optimal balance between L_1 and L_2 is expected to change substantially.

Our model outperforms both Bracken and DiTASiC for every error measure and every taxonomic rank, which makes us confident that the linear model is a better approximation for this problem compared to the Bayesian re-estimation performed by Bracken, and that the regularization effects are beneficial since they provide a satisfying variable selection compared to both Bracken and DiTASiC, the latter including basically half of the index in the output (648 FP over 1267 genomes in the index). Nevertheless, real metagenomic samples introduce challenges which may break the assumptions needed for the linear model to converge and provide meaningful estimations. While testing these conditions is not straightforward, since results on real samples may only be compared to those of other state-of-the-art tools which are not necessarily more accurate, we use a more realistic, ad-hoc simulation experiment to account for some of these challenges. Furthermore, we test our model on what is, to our knowledge, one of few metagenomic datasets for which the relative abundances of each component is known. We will see that even in this case it is hard to assess which part of the error is due to the DNA extraction protocol and the read sequencing process.

4.1.2 Human Microbiome Project Mock Community

In a real metagenomic scenario, where samples are collected directly from the environment and sequenced without prior isolation, it is extremely important to use

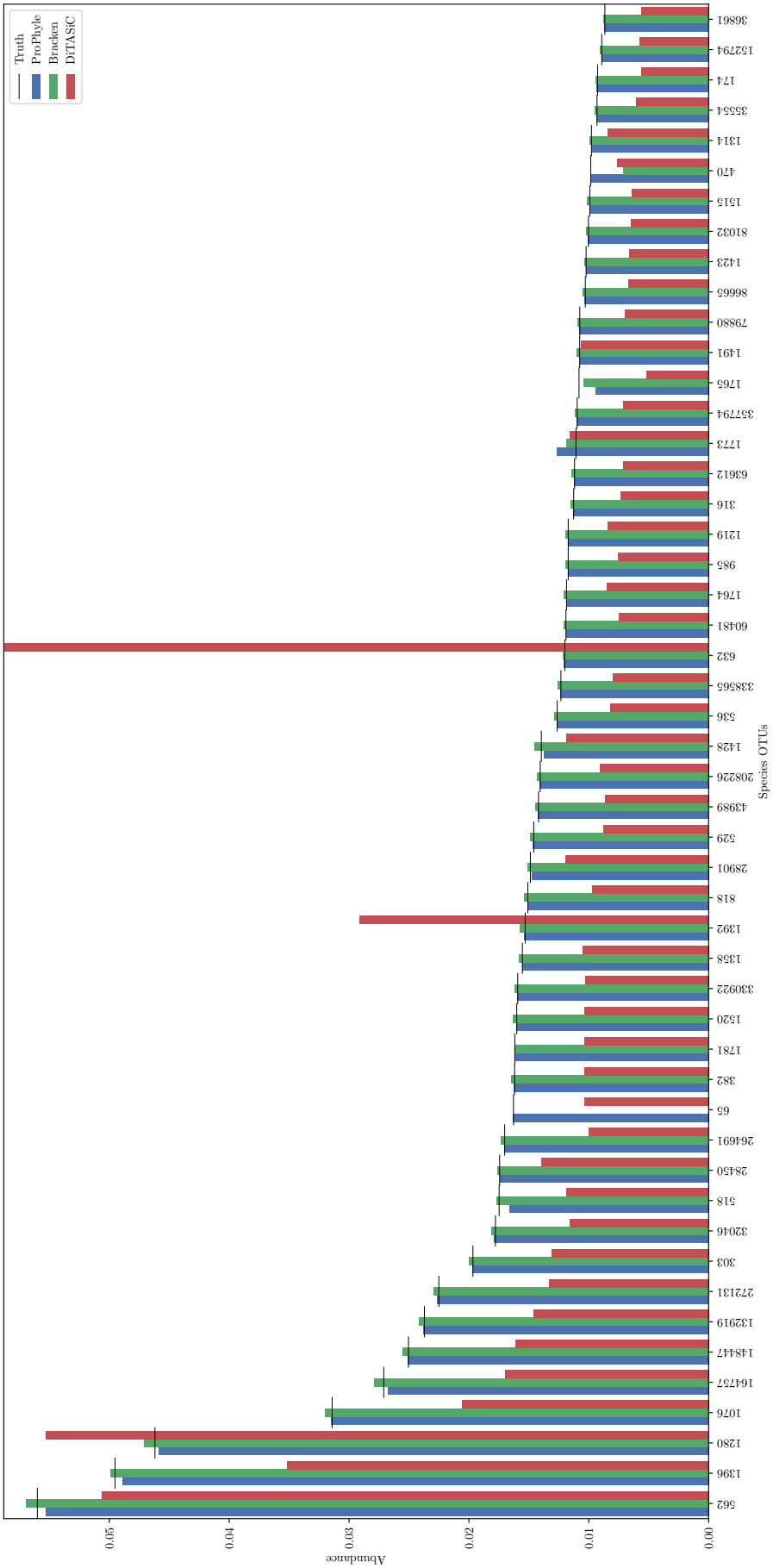


FIGURE 4.1: Abundance profiles at the species level estimated by ProPhyle, Bracken and DiTASiC for the simulated dataset from [24].

an extensive database of reference genomes to reduce the number of reads which cannot be classified for lack of a close-enough representative genome. This has recently been explored in [8], where the accuracy of the pipeline Kraken+Bracken was assessed for several versions of NCBI RefSeq, one of the most commonly used reference databases. For this reason, we have used the last version of RefSeq as of July 2018 to build the index of ProPhyle. Due to the size of the database, for which the construction of Kraken's index requires about 2.5TB of RAM and 11 days on a 64 cores compute node [8], we have selected only the *reference* and *representative* genomes for each species, as defined in their website[1]. This selection is mantained by NCBI to provide the highest diversity and quality of reference genomes. The total size of the reference database amounts to 30GB, and the resulting ProPhyle index requires 55GB of memory. Unfortunately, even with this reduced database it was unfeasible to run Kraken, since its index was estimated to require 330GB of memory.

In addition to this issue, real metagenomic samples pose a great challenge to the performance benchmarks of classifiers, since the actual abundances in the sample are often unknown. To address this issue, we have used one of the few readsets for which a ground truth is provided: the pilot experiment for the Human Microbiome Project (HMP). The Human Microbiome Project is a research initiative to improve the understanding of the microbial flora involved in human health and disease, and in this pilot experiment they designed an even mixture of microbial DNA for which at least one reference was available in RefSeq at the time of the experiment. This provides a semi-realistic scenario for the benchmark of abundance estimation: while the data has been obtained by sequencing DNA with a physical machine, the composition of the sample is far from being realistic, since it only includes 22 organisms with uniformly distributed abundances.

Since some of the organisms which were used for the experiment have been removed from RefSeq, we only analyzed the performance at ranks species and higher. This poses another challenge to classification, since only a close relative genome is available in the reference database.

Rank	Measure	Uniform	Adjusted
Genome	RSS	0.15	0.03
	FN # (ab.)	3 (0.14)	3 (2.28e-6)
	FP # (ab.)	5 (0.01)	5 (0.01)
Species	RSS	0.15	0.03
	FN # (ab.)	1 (0.05)	1 (7.61e-7)
	FP # (ab.)	3 (6.30e-3)	3 (6.30e-3)
Genus	RSS	0.15	0.03
	FN # (ab.)	0 (0)	0 (0)
	FP # (ab.)	1 (5.66e-3)	1 (5.66e-3)

Comparing the results of ProPhyle to the uniform distribution which we were expecting showed that there was a considerable difference between the estimated and real abundances. Therefore we used NCBI's official analysis results as ground truth, and added the adjusted error estimates in the last column. Those abundances resembled very closely the ones estimated by ProPhyle, and this shows that there may be some algorithmic issue common to several computational methods. This is probably linked to contaminations in the database, since there was one genome in particular which was assigned 40% of the reads. We expect this kind of issues to

[1] https://support.nlm.nih.gov/knowledgebase/article/KA-03578/en-us

happen frequently in real experiment, and there is a real need for a reliable reference database which can limit these discrepancies.

4.1.3 Stress test

In this experiment we simulated very complex metagenomes with low coverage from many different species. 10 samples were generated, each containing 500k read pairs of length 150bp originated from 250 different reference genomes. Since the abundance profile fits an exponential distribution, 50 least abundant species (<e-5) of each sample only have 5 or less representative reads, and have been used to account for noise in the data (reads which originate from organisms not in the index, but which may match other relatives). Also for this experiment we chose NCBI's selection of reference and representative genomes used in 4.1.2.

The purpose of this experiment was to show how different choices of parameters can fit metagenomes of different complexity, and we achieved error profiles similar to 4.1.1 while successfully filtering out the low abundant organisms used for the simulation of noise.

4.2 Discussion

The experiments in the previous section show that the model is well defined and provides results comparable to state-of-the-art abundance estimators. Furthermore, compared to the work which inspired this[13], the approach suggested is scalable to reference databases containing several thousands of genomes, as shown in 4.1.2. This is possible thanks to the properties of the metagenomic classifier ProPhyle which produces accurate assignments in a fraction of the time required by traditional read alignment methods as those included in [13].

The parametrization of the regularized linear model makes the method extremely flexible. As shown in 4.1.1, by using a LASSO configuration it is possible to roughly estimate the composition of a sample, with virtually perfect results starting already at the species level; by introducing the L_2 penalty in the model and reducing the regularization weight, the method predicts extremely accurate abundances even at the genome level, at the price of introducing some false positives in the results.

One issue we find important to address is the heteroskedastic nature of assignment counts, i.e. the fact that different values of the response variable have different variance in their errors. This is not taken into account in the current model, since standard linear regression implies an homoscedasticity assumption. Although we tried to address this problem by log-transforming the response variables and using a Poisson GLM with log link, we found that both the error estimates and the variable selection properties were only worsened. We believe this is due to a numerical convergence issue, triggered by the high frequency of 0s in both the assignment counts vector and in the similarity matrix.

The exponential growth of the reference databases also poses a threat to the scalability of the method, since simulating and assigning reads from the whole database may take weeks of computation in the near future. As we show in 4.1.2, the method can still provide useful insights on the data when only few representatives for each species are included in the database, even if the genomes in the sample are not included.

Chapter 5

Conclusions

In this work we analyze the flaws in state-of-the-art abundance estimators for metagenomics, and we propose a method based on linear regression to correct the biases linked to the similarity of genomes in the reference database. Using the Elastic Net method for regularization, we obtain enhanced selection of the true positives in the sample with respect to other popular tools, while also improving the error estimates. We believe this method has a wide range of applications, and we provide its implementation in the open-source metagenomic classifier ProPhyle.

It is of particular biological interest that this method provides extremely flexible abundance estimation capabilities: by tuning the regularization parameters, it is possible to match the complexity of different metagenomic experiments in real time, without the need to re-run costly simulation or learning steps. This is extremely useful in situations where the compositions of the samples are completely unknown, since it enables their analysis using an extensive database of reference genomes and provides accurate estimates of the abundances of each taxonomic clade within minutes, as well as samples of well-known expected composition and complexity, where the task is to accurately estimate genome-level abundances to compare them to other samples.

Nevertheless, there are some challenges still to be addressed: first, genomes present in very low abundances are very difficult to detect, since their coefficients are likely to be annihilated during the optimization of the regularized model. On the other hand, reducing the regularization parameters may have an even worse effect on the results, introducing hundreds of low-abundant false positives indistinguishable from those actually present in low abundances. This issue is intrinsic to the nature of microbial genomes, subject to phenomena such as horizontal gene transfer thanks to which prokaryotic organisms of different species can exchange portions of their genetic material, decreasing the probability that reads can be assigned to the exact reference genome they were generated from, in the current state of read sequencing technologies, producing short reads which may map equally well to tens or hundreds of genomes, and in the heuristic assignment algorithms which are unavoidable if we want to make the analysis of metagenomes computationally feasible. The scientific literature confirms that it is unlikely to obtain accurate genome-level abundances for real-sized metagenomic samples and reference datasets [8], [14], [25], without restricting the search to a tractable number of "genomes of interest".

The field of metagenomics has an incredible potential for uncovering the role of microbial communities in our life and our ecosystems. We need more efficient algorithms to analyze these noisy, complex, high-dimensional data and understand how these tiny, but numerous organisms are implied in the biological mechanisms of our species. We hope that this work will be a small step towards a better understanding of these data.

Bibliography

[1] J. Lu, F. P. Breitwieser, P. Thielen, and S. L. Salzberg, "Bracken: Estimating species abundance in metagenomics data", en, *PeerJ Computer Science*, vol. 3, e104, Jan. 2017, ISSN: 2376-5992. DOI: 10.7717/peerj-cs.104. [Online]. Available: https://peerj.com/articles/cs-104.

[2] G. Kucherov, "Algorithms for biosequence search: Past, present and future", *arXiv:1808.01038 [q-bio]*, Aug. 2018, arXiv: 1808.01038. [Online]. Available: http://arxiv.org/abs/1808.01038.

[3] K. Brinda, "Novel computational techniques for mapping and classification of Next-Generation Sequencing data", Theses, Université Paris-Est, Nov. 2016. [Online]. Available: https://hal.archives-ouvertes.fr/tel-01484198.

[4] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool", eng, *Journal of Molecular Biology*, vol. 215, no. 3, pp. 403–410, Oct. 1990, ISSN: 0022-2836. DOI: 10.1016/S0022-2836(05)80360-2.

[5] D. H. Huson, A. F. Auch, J. Qi, and S. C. Schuster, "MEGAN analysis of metagenomic data", en, *Genome Research*, vol. 17, no. 3, pp. 000–000, Jan. 2007, ISSN: 1088-9051, 1549-5469. DOI: 10.1101/gr.5969107. [Online]. Available: http://genome.cshlp.org/content/early/2007/01/01/gr.5969107.

[6] H. Li, "Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM", *arXiv:1303.3997 [q-bio]*, Mar. 2013, arXiv: 1303.3997. [Online]. Available: http://arxiv.org/abs/1303.3997.

[7] D. E. Wood and S. L. Salzberg, "Kraken: Ultrafast metagenomic sequence classification using exact alignments", *Genome Biology*, vol. 15, no. 3, R46, Mar. 2014, ISSN: 1474-760X. DOI: 10.1186/gb-2014-15-3-r46. [Online]. Available: https://doi.org/10.1186/gb-2014-15-3-r46.

[8] D. J. Nasko, S. Koren, A. M. Phillippy, and T. J. Treangen, "RefSeq database growth influences the accuracy of k-mer-based species identification", en, *bioRxiv*, p. 304 972, Apr. 2018. DOI: 10.1101/304972. [Online]. Available: https://www.biorxiv.org/content/early/2018/04/19/304972.

[9] K. Břinda, K. Salikhov, S. Pignotti, and G. Kucherov, *ProPhyle: A phylogeny-based metagenomic classifier using the Burrows-Wheeler Transform*, eng, Jul. 2017. DOI: 10.5281/zenodo.1045427. [Online]. Available: https://zenodo.org/record/1045427.

[10] N. L. Bray, H. Pimentel, P. Melsted, and L. Pachter, "Near-optimal probabilistic RNA-seq quantification", en, *Nature Biotechnology*, vol. 34, no. 5, pp. 525–527, May 2016, ISSN: 1546-1696. DOI: 10.1038/nbt.3519. [Online]. Available: https://www.nature.com/articles/nbt.3519.

[11] D. Kim, L. Song, F. P. Breitwieser, and S. L. Salzberg, "Centrifuge: Rapid and sensitive classification of metagenomic sequences", en, *Genome Research*, Oct. 2016, ISSN: 1088-9051, 1549-5469. DOI: 10.1101/gr.210641.116. [Online]. Available: http://genome.cshlp.org/content/early/2016/11/16/gr.210641.116.

[12] A. Dilthey, C. Jain, S. Koren, and A. Phillippy, "MetaMaps - Strain-level metagenomic assignment and compositional estimation for long reads", en, *bioRxiv*, p. 372 474, Jul. 2018. DOI: 10.1101/372474. [Online]. Available: https://www.biorxiv.org/content/early/2018/07/20/372474.

[13] M. S. Lindner and B. Y. Renard, "Metagenomic abundance estimation and diagnostic testing on species level", en, *Nucleic Acids Research*, vol. 41, no. 1, e10–e10, Jan. 2013, ISSN: 0305-1048. DOI: 10.1093/nar/gks803. [Online]. Available: https://academic.oup.com/nar/article/41/1/e10/1164154.

[14] M. Fischer, B. Strauch, and B. Y. Renard, "Abundance estimation and differential testing on strain level in metagenomics data", en, *Bioinformatics*, vol. 33, no. 14, pp. i124–i132, Jul. 2017, ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btx237. [Online]. Available: https://academic.oup.com/bioinformatics/article/33/14/i124/3953953.

[15] T. Garnier, K. Eiglmeier, J.-C. Camus, N. Medina, H. Mansoor, M. Pryor, S. Duthoy, S. Grondin, C. Lacroix, C. Monsempe, S. Simon, B. Harris, R. Atkin, J. Doggett, R. Mayes, L. Keating, P. R. Wheeler, J. Parkhill, B. G. Barrell, S. T. Cole, S. V. Gordon, and R. G. Hewinson, "The complete genome sequence of Mycobacterium bovis", en, *Proceedings of the National Academy of Sciences*, vol. 100, no. 13, pp. 7877–7882, Jun. 2003, ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.1130426100. [Online]. Available: http://www.pnas.org/content/100/13/7877.

[16] B. Langmead and S. L. Salzberg, "Fast gapped-read alignment with Bowtie 2", en, *Nature Methods*, vol. 9, no. 4, pp. 357–359, Apr. 2012, ISSN: 1548-7105. DOI: 10.1038/nmeth.1923. [Online]. Available: http://www.nature.com/articles/nmeth.1923.

[17] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso", *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996, ISSN: 0035-9246. [Online]. Available: https://www.jstor.org/stable/2346178.

[18] *Linear regression*, en, Page Version ID: 851432903, Jul. 2018. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Linear_regression&oldid=851432903.

[19] D. Donoho, "Compressed sensing", en, *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006, ISSN: 0018-9448. DOI: 10.1109/TIT.2006.871582. [Online]. Available: http://ieeexplore.ieee.org/document/1614066/.

[20] *Lasso and Elastic Net — scikit-learn 0.19.2 documentation*. [Online]. Available: http://scikit-learn.org/stable/auto_examples/linear_model/plot_lasso_coordinate_descent_path.html (visited on 08/23/2018).

[21] L. Breiman, "Heuristics of instability and stabilization in model selection", en, *The Annals of Statistics*, vol. 24, no. 6, pp. 2350–2383, Dec. 1996, ISSN: 0090-5364, 2168-8966. DOI: 10.1214/aos/1032181158. [Online]. Available: https://projecteuclid.org/euclid.aos/1032181158.

[22] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net", en, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, pp. 301–320, Apr. 2005, ISSN: 1467-9868. DOI: 10.1111/j.1467-9868.2005.00503.x@10.1111/(ISSN)1467-9868.TOP_SERIES_B_RESEARCH. [Online]. Available: https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2005.00503.x%4010.1111/%28ISSN%291467-9868.TOP_SERIES_B_RESEARCH.

[23] K. Břinda, V. Boeva, and G. Kucherov, "RNF: A general framework to evaluate NGS read mappers", en, *Bioinformatics*, vol. 32, no. 1, pp. 136–139, Jan. 2016, ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btv524. [Online]. Available: https://academic.oup.com/bioinformatics/article/32/1/136/1742858.

[24] D. R. Mende, A. S. Waller, S. Sunagawa, A. I. Järvelin, M. M. Chan, M. Arumugam, J. Raes, and P. Bork, "Assessment of Metagenomic Assembly Using Simulated Next Generation Sequencing Data", en, *PLOS ONE*, vol. 7, no. 2, e31386, Feb. 2012, ISSN: 1932-6203. DOI: 10.1371/journal.pone.0031386. [Online]. Available: http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0031386.

[25] S. Lindgreen, K. L. Adair, and P. P. Gardner, "An evaluation of the accuracy and speed of metagenome analysis tools", en, *Scientific Reports*, vol. 6, p. 19 233, Jan. 2016, ISSN: 2045-2322. DOI: 10.1038/srep19233. [Online]. Available: https://www.nature.com/articles/srep19233.