Benchmarking Machine Learning performances with Microbiome Data

Master's thesis
in the single-subject master program, subject Biology
of the Faculty of Mathematics and Natural Sciences
of the Christian-Albrechts-University of Kiel

submitted by
Jennifer Neumaier

First examiner: Dr. Elin Org
Second examiner: Prof. Dr. Tal Dagan

Kiel, June 2022

1 TABLE OF CONTENTS

2 ZUSAMMENFASSUNG

Machine Learning wird in Mikrobiomstudien bereits viel genutzt und das Interesse daran ist weiterhin stetig am Wachsen. Problematisch daran ist allerdigs, dass es noch kein universelles Verständnis über die genaue Funktionsweise der Algorithmen gibt, gerade im Bezug zu Mikrobiomdaten. Deswegen ist es ein interessantes und viel diskutiertes Thema das einen großen Einfluss hat auf mögliche Anwendungen mit Mikrobiomdaten. Ein Schlüsselthema in der Mikrobiomforschung ist der mathematische Raum, in dem die Mikrobiomdaten existieren. Sequenzierdaten erscheinen als reine Zähldaten, in Wahrheit sind sie aber nur relative Häufigkeiten, in der Mathematik auch „Compositional Data" genannt. Das hat zur Folge, dass die Daten erst transformiert werden müssen, bevor man sie nutzen kann. Das bringt mit sich gewisse Limitationen, die einen Einfluss auf die Leistung und Aussagekraft von Machine Learning Modellen haben können. Log-ratio Transformationen sind ein Vorschlag, der schon von einigen Studien vorgebracht wurde und bereits in der statistischen Analyse von Mikrobiomdaten genutzt wird. Ihr Einfluss auf Machine Learning Modelle wurde allerdigs nie in größer angelegten Studien untersucht.

In diesem Projekt werden deshalb verschiedene log-ratio Transformationen auf verschiedene Mikrobiomdatensätze angewandt. Diese Datensätze besitzen verschiedene Charakteristika, wie Datensatzgröße oder Korrelation der Daten. Diese log-ratio transformierten Daten werden dann in Machine Learning Modelle gegeben und ihre Leistung verglichen zu *CoDaCoRe,* einem Algorithmus der spezifisch für Mikrobiomanalysen erstellt wurde.

Die Ergebnisse zeigen, dass log-ratio Transformationen die Vorhersagefähigkeit von Machine Learning Modellen erhöhen, wenn man es zu Datensätzen vergleicht, die nur standardmäßig normalisiert wurden. Die stärkere Leistung ist auch im Vergleich zu *CoDaCoRe* zu finden*.* Weitere Tests in diesem Projekt zeigten auch, dass log-ratio Transformationen vermutlich keine weiteren systematischen Fehler in Machine Learning Workflows eintragen und dass eher Arbeitsprozesse wie Imputationen und Filtern von Daten noch genauer untersucht werden müssen, um stabile Modelle mit Mikrobiomdaten zu bekommen.

3 ABSTRACT

Machine Learning in microbiome studies is widely used and the interest is growing. However, there is no universal understanding of the algorithmic approaches that can best utilize the information present in the microbiome data. Thus, this is an interesting and widely discussed topic that can have a great impact on the potential applications leveraging microbiome data. A key topic in microbiome research is the sample space of the input data. The sequencing data appears as count data, but, only relative abundance of the microbial features can be observed, commonly called "compositional data". Thus, transforming the read counts to relative abundances is usually the first step and Machine Learning methods are mostly applied on relative abundances. However, relative abundances raise several limitations, which can have an impact on the performance of the prediction models. Therefore, log-ratio transformations are a proposition made by several studies now, although their impact on Machine Learning performances has never been tested in large-scale studies. In this benchmarking project several log-ratio transformations have been applied to microbiome data sets with various characteristics. Afterwards, performances of Machine Learning models have been assessed and compared to *CoDaCoRe*, an algorithm specifically made with microbiome analysis in mind.

The results show that applying log-transformations before applying Machine Learning models increases performances compared to standard normalization techniques and also *CoDaCoRe*. Further tests also indicate that log-ratio transformations introduce no additional biases in Machine Learning pipelines and problems like data leakage during imputation and filtering have to be assessed further to ensure stable models with microbiome data.

4 INTRODUCTION

Working with mathematical concepts is always a bit out of the comfort zone for most biologists. The field of statistics looms like a big black cave which is only lightened in a few places, but we have to traverse it anyways. Unfortunately, with Machine Learning encroaching also into biological research questions, this cave just became much bigger, without really bringing any additional light.

Machine Learning provides a useful tool to exploit information from biological data sets. It brings the possibility of predicting host-phenotypes, finding associations between features, and using microbial communities to characterize patients. Several papers (Jasner et al. 2021; Kubinski et al. 2022; Coenders and Greenacre 2021; Zhang and Shi 2019) already gave some practical advice on how to use Machine Learning with microbiome data, however a vital concept concerning the mathematical characteristics of microbiome data has been almost completely ignored in the last few years.

It has been made abundantly clear in the last years that sequencing data is of compositional nature, which means it has distinct mathematical characteristics than other data types (Greenacre et al. 2021b; Gloor et al. 2017; Quinn et al. 2018). While some papers already confirmed that log-ratio transformations do increase Machine Learning performances (Jasner et al. 2021; Kubinski et al. 2022; Quinn and Erb 2020; Zhang and Shi 2019), it has never been tried to assess how other important parameters (like sample sizes, feature sizes or test set methodologies) might be affected by the application of log-ratio transformations in Machine Learning models and whether or not these parameters demand a choice in log-ratio transformation.

The goal of this master thesis is to look a bit deeper into the combination of log-ratio transformations and Machine Learning models. For that, the properties of compositional data will be introduced, and the idea of log-ratio transformations explained. Afterwards, Machine Learning performances will be compared more in detail with changing parameters to assess if log-ratio transformations introduce new difficulties in Machine Learning models.

As this master thesis uses microbiome sequencing data and was created in a microbiome research group, this text will mostly focus on this field and its papers. All results should be applicable to other high-throughput sequencing data, as well as any data that is in some way confined by an arbitrary sum. Such data is found for example in geochemistry, ecology, sociology, political sciences, etc., and therefore ultimately spans the problematic into various different fields (Greenacre et al. 2021a).

4.1 CHARACTERISTICS OF COMPOSITIONAL DATA

In order to define and illustrate the concept of compositional data, let’s assume a classic biological example. The following Figure (Fig.1A) shows two different ecological fields: A and B. In field A, four rabbits, seven birds, eight bees and one wolf have been counted, whereas field B contains two rabbits, four birds, four bees and one wolf. It becomes clear that, as similar as the diversity may be, field B seems to have only half of the population of field A. This in itself is already valuable information. The total counts per field can be easily preserved in our data collection and therefore, the absolute count of each organism in this field matters.

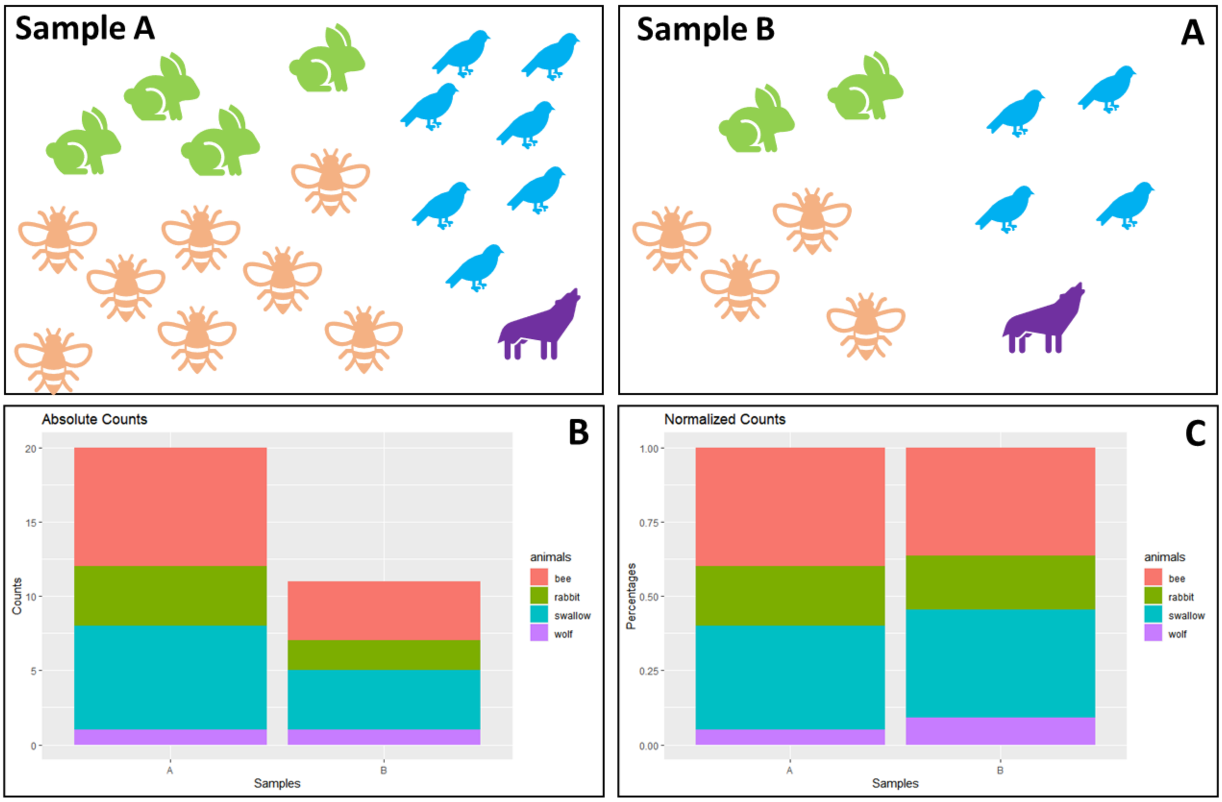

Figure 1: Information Loss of Normalized Data

(A) Illustration of the number of animals found in two different samples. Field A contains four rabbits, eight bees, seven birds and one wolf, whereas field B contains two rabbits, four bees, four birds and one wolf. In (B) the absolute counts have been plotted as a stacked bar plot, with each animal in a different color. (C) shows the stacked bar plot as normalized counts, e.g., here percentages.

When using absolute counts, the difference between both fields is easily visible (Fig.1B). However, when we really want to compare both fields, we need to transform the samples to a common scale. This is called normalization and we can see the effect in (Fig.1C). As soon as the data is normalized, the particular information of absolute counts gets lost. When collecting ecological data ourselves, we can preserve the fact that field B only contained eleven individuals and field A contained twenty, by saving that number somewhere in an Excel sheet. However, the problem with sequencing data is: we get the data in the form of Fig.1C.

To demonstrate how a sequencing machine cannot preserve absolute counts, imagine the following situation: We want to sample field A multiple times a day, but in order to be more efficient, we buy a machine to do the counting for us. Three times a day this machine transmits the number of all the different animals coming to this field. However, this machine has one flaw: it can only count to twenty. As soon as the twenty-first animal on this day comes to the field, it is just simply not counted.

Consequently, the overall number twenty carries no meaning. Every sample has this exact total number, so it carries no valuable information. Of course, a limit of twenty is weird for us to understand, but sequencing machines do a very similar thing. They are limited in their capacity on e.g., the flow cells and even the most effective sequencing machines could never fully sequence the entirety of the organism's DNA contents. Not only the technology limits our counting capabilities, but the sampling procedures to acquire microbiome data can only entail a small fraction of the actual bacterial content. Thus, the total number of sequences measured by sequencing machines ultimately depends on the chemistry of the assay, not the input material (Quinn et al. 2018).

The consequence of this sampling problem is, that we have to accept the fact that the sum of counts in sequencing data are irrelevant. This is particularly problematic as we cannot compare groups of data as we are used to do it. A lot of tools in microbiome data analysis stem from the field of ecology and contain methods like richness or diversity to characterize samples. If we only had the information of Fig.1C, then we would perhaps assume that field A and B share the same ecology. Or perhaps, if this were microbiome data, we would maybe falsely assume that these samples came from patients of the same group.

Data that is constraint in that way is called "Compositional Data". It was first introduced by John Aitchison, roughly 40 years ago and since then steadily improved and further developed. Thanks to this mathematical theory, we can still draw information from sequencing data. We just have to adjust for the fact that the absolute counts are non-informative (Quinn et al. 2018; Greenacre et al. 2021b; Aitchison 1982). We can instead use relative abundances, sometimes also called proportions, between features in a sample.

At first glance, this does not sound as if it would have serious consequences. However, in the next section we will look at the mathematical concepts behind compositional data and its properties and demonstrate, that the concept of compositional data changes the whole standard downstream analysis (Gloor et al. 2017).

4.2 THE SIMPLEX SPACE

As mentioned, absolute counts in compositional data are unavailable and only relative abundances are of interest. This puts the data in the so-called “Simplex Space”, instead of the more familiar Euclidean space. The following Figure 2 shows how the data from field A in section 2.1 would look like in the Simplex Space:

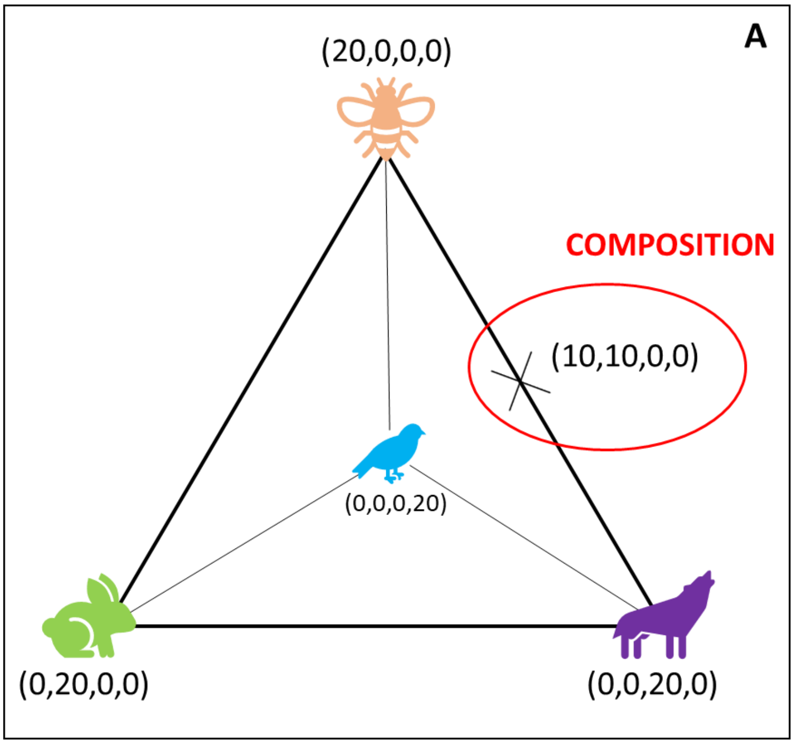

Figure 2: Biological Example in the Simplex Space

Assuming the collected ecological data from Figure 1 is compositional, it can be visualized in a Simplex space. Geometrically, a tetrahedron is created with all different components (here animals) placed on the four corners of a polytope. A composition is one possible combination of components confined in the Simplex Space.

We stick with the ecological example and place all animals as one corner in a geometrical space. With four features, we are able to create a geometric figure called tetrahedron that represents the Simplex Space visually.

We use the flawed machine, and at one day, we sample twenty rabbits in field A. This would lead to a point in the simplex space that sits directly in the left corner, with the coordinates (0,20,0,0), because we only have rabbits, no other animal. Another day, we sample only twenty bees, and no other animals, then we would find our data point where the bee is, at the very top. Marked in red is a sample where the machine counted ten bees, ten wolves and no birds and no rabbits. Every sample round produces one “composition” and the examples show, that the distance between any two variables is sensitive to the presence or absence of other components (Quinn et al. 2018). If a composition is moved from one corner of the animal-simplex, it means that the other values in the composition are directly changed. Consequently, that makes all variables *mutually dependent* on one another. In literature this data is also called "spurious" because it appears as if the data points have a causal relationship when the perceived correlation is only due to data properties. When a composition is moved from bees in the direction of wolves it seems like there is a causal relationship because the increase in the number of wolves, directly decreases the number of bees.

To describe this a bit more mathematically, the problem described above is formally known as "the negative bias problem" (other names are also the constant-sum problem, the closure problem, or the null correlation difficulty) (Aitchison 2003). It means, that negative covariances are presupposed by the limitation of the sum, instead of produced by stochastic factors (Pawlowsky-Glahn and Egozcue 2016; Aitchison 2003). Covariances are similar to correlation coefficients, only that the latter is constraint between -1 and 1, but otherwise they can be interpreted similarly. Translated, a negative covariance means that two factors are negatively correlated, i.e., if one factor goes up, another goes down. Exactly as has been demonstrated by the ecological example before.

Thus, using any form of statistical test or machine learning tool seems redundant, as errors in correlations, univariate and multivariate tests are almost preconditioned, and we easily would make false assumptions about the correlation of the data. A correct handling of compositional data and the Simplex Space is therefore not optional (Gloor et al. 2017).

4.3 MAPPING THE SIMPLEX SPACE INTO EUCLIDEAN SPACE

In the ecology sample, the Simplex could have been easily overcome by e.g., normalizing to a field size. This indirectly preserves information about the total number of animals. Similarly, it has been tried for sequencing data to calculate an “effective library size” and therefore recover this way the original scale of data. For that, normalization methods like trimmed mean of M-values (TMM) have been introduced, as well as Reads per Kilobase Million (RPKM) and Transcripts per Kilobase Million (TPM) (Quinn et al. 2018). However, all of those methods involve rescaling counts by the library size and these normalizations come with the drawback that some of these methods are sensitive to the removal of low abundant counts, as well as to data symmetry (Quinn et al. 2018).

Furthermore, Aitchison already criticized very early that there is no “magic to open up closed data” (Aitchison 2003), which is what normalization tries to do. Moreover, since information provided in compositional data is essentially about ratios of the components, it seems logical to also think in terms of ratios. Methods that are able to deal with the Simplex Space when analyzing compositional were introduced by Mateu-Figueras et al., (2011) with the “staying-in-the-simplex” approach or Greenacres (2017) “pragmatic approach”. However, these methods require a technical understanding of the algebraic-geometric structure of the Simplex and are therefore relatively complex.

No, the best way forward in Aitchison’s mind was to transform the data in a way that allows us to use it with Euclidean Space rules. The general idea is, that the Simplex Space is endowed with a Euclidean space structure. This has several mathematical advantages: if one can map the Simplex Space into Euclidean Space, then we can use all the to us more familiar methods and don’t have to rely on more complicated equations to analyze data (Pawlowsky-Glahn and Egozcue 2016). Aitchison (1982) proposal to achieve that was to use log-ratio transformations.

Here, the focus will be on these log-ratio transformations, as they have been more heavily favored in the last decades due to their practicability (Greenacre et al. 2022). They also come with the advantage that they can be easily added to pre-existing Machine Learning pipelines that have been proposed and created over the last years. The next section introduces them more in detail.

4.4 LOG-RATIO TRANSFORMATIONS

When Aitchison (1982) first tried to overcome the limitations set by compositional data he defined some principals to lay down as fundamentals to the good practice of compositional data. These are important as they paved the way that data could be mapped into Euclidean space correctly.

The most agreed upon properties are scale invariance (compositions do not change with e.g., sequencing depth), perturbation invariance (i.e., converting a composition between equivalent units will not change the results), and permutation invariance (i.e., changing the order of the components within a composition will not change the results) (Quinn et al. 2018; Greenacre et al. 2022)

It's clear why these concepts are commonly accepted. Imagine a composition of the ecology example with composition (8,8,2,2), measuring eight bees, eight wolves, two swallows and two rabbits. Scale invariance means that a transformation should ensure that we get the same ratios even if we had a machine that was able to count to forty instead of twenty and when we would get the composition (16,16,4,4), the ratios between the components are stable. Similarly, if we would not count to the base of ten, but rather to the base of hundred, the transformation should not change the ratios. Lastly, the ratios should not be affected even if we sampled data in another order (e.g., two rabbits, eight wolves, two swallows, eight bees).

Three more properties exist that are more heavily discussed in their importance and if they are strictly necessary: isometry (i.e., the distances between ratios should be exactly the same after mapping into Euclidean space) and sub-compositional coherence (i.e., overlapping components in two compositions of the same measurement should have the same ratios) (Quinn et al. 2018; Greenacre et al. 2021a; Greenacre et al. 2022).

Imagine two different scientists observed the same field but scientist A finds two birds, two rabbits, eight bees and eight wolves, whereas scientist B finds two birds, two rabbits and eight wolves. Scientist B was not able to find one animal from the original composition and if one would normalize the results to 1, both scientists would get completely different result (Scientist A: (0.4,0.1,0.1,0,4) and Scientist B: (0.16,0.6,0.16)), respectively. However, if ratios are studied, the overlapping components give the same ratios (e.g., the ratio birds/rabbits in both cases is 1). Given the same example the idea of isometry can also be explained. The distance between eight wolves and two rabbits in the raw compositional data of scientist A is 8-2 = 6, whereas in a log-transformed data set the distance is ln(8)-ln(2) = 1.3. This would mean that in this example applying the natural log does not conform to isometry, as 1.3 ≠ 6 and some advocate that these distances should be preserved in log-ratio transformations.

One could argue that all mentioned properties are important, unfortunately adhering to all of them has practical implications. Aitchison proposed log-transformations, because they already adhere to most of the properties inherently, like scale variance, perturbation invariance and permutation invariance. However, properties like isometry and sub-compositional coherence are more difficult.

The log-ratio transformation that imparts all those properties is called isometric log-ratio (ILR). ILRs are considered the gold standard of log-ratio transformations by some experts, as they engender exactly the same multivariate geometric structure of the sample points as that of the formerly mentioned Aitchison geometry (Greenacre et al. 2021b). ILR map a composition in the D-part Aitchison-Simplex isometrically to a D-1 dimensional Euclidian vector, and the exact procedure is quite difficult to understand (Greenacre et al. 2021a; Greenacre et al. 2022). Additionally, ILRs are particularly problematic when the numbers of components are high as the computational power to calculate ILRs increases significantly (Greenacre et al. 2021b). Considering that microbiome data is usually very high-dimensional would make ILR already an inadequate solution.

Thankfully, there are more types of log-ratio transformation, that are easier to understand and therefore interpret. As a result of this, it was decided to use two types of log-ratio transformation here: ALR (additive log-ratio) and CLR (centered log-ratio)

$$\begin{aligned}\mathrm{CLR}(j) &= \log\left(\frac{x_j}{g(x)}\right), \quad j = 1, \ldots, D \\ &= \log(x_j) - \frac{1}{D}\sum_k \log(x_k)\end{aligned}$$

Figure 3: Equation for CLR

The equation describes the calculation of CLR, with x_j as vector of sample features, D_j the total number of features, and g(x) the geometric mean of sample vector x. Log-ratio transformations are applied within a sample (i.e., row-wise).

In CLR, log-ratios are computed between each component and the geometric mean of all components (Gloor et al. 2017). It has the advantage that it is computationally easy to do compared to ILRs. Furthermore, it reproduces the log-ratio geometry perfectly. However, it is not sub-compositionally coherent, because the whole composition (i.e., sample) is used to calculate the geometric mean and every sample will therefore use a different geometric mean, as well as every sub-composition. Unfortunately, it is not very useful in sparse data containing a lot of zeroes as this can skew the geometric mean (Gloor et al. 2017).

The second log-ratio transformation is ALR. Here, the log-ratio is taken of each feature within a composition and divided by a chosen reference feature.

$$\begin{aligned}\mathrm{ALR}(\,j \,|\, ref\,) &= \log(x_j/x_{ref}),\; j = 1, \ldots, D,\; j \neq ref \\ &= \log\,(x_j) - \log\,(x_{ref})\end{aligned}$$

Figure 4: Equation ALR

The equation describes the calculation of ALR, with x_j as vector of sample features, D the total number of features, and x_{ref} the reference feature. Log-ratio transformations are applied within a feature (i.e., column-wise).

Thus, the interpretation of ALR log-ratios is very straight-forward, as one can interpret a change in all features with respect to the reference. ALR transformed values are also sub-compositional coherent, which is traded for a small loss in isometry. The biggest problem with ALR is the choice of reference. A reference can simply be chosen by the scientist, for example when they want to compare one bacterium to all others. However, if one is unsure about the choice of reference, Greenacre et al. (2021) proposed to use three criteria to find a good reference: (i) the reference component should maximize the Procrustes correlation between the additive log-ratio geometry and the exact log-ratio geometry, (ii) the reference should minimize the variance of log-transformed components, and (iii) it should be a well-populated component.

This means the optimal reference can be determined computationally. Using these guidelines produces additive log-ratios being near-isometric, which would make them a favorable log-transformation. The obvious drawback is the computational complexity when a Procrustes analysis is used, which increases especially in higher-dimensional data.

4.5 THE PROBLEMS

Machine Learning models are of great interest for microbiome analysis, as they allow detection of biomarkers, phenotypes or microbial taxa, as well as other interesting tasks, that are not possible with the standard microbiome tool kit (Marcos-Zambrano et al. 2021). In supervised Machine Learning a model is learned that makes predictions on new data and can identify genomic features that underlie a predictive model (Whalen et al. 2022). The hopes are that with the help of AI-based techniques we can transform clinical decision-making and extract more knowledge from the amount of data available to us (Quinn et al. 2022).

However, while modern machine learning algorithms are very powerful, they come with a great lack of transparency. It is hard to understand how and why the model makes predictions. Especially deep neural networks are considered “black boxes” as the inner workings remain incomprehensible to the outsider and the lack of quality assurances goes against good scientific practice (Quinn et al. 2022). As it has been shown that well-specified machine learning models outperform deep learning models (Bailly et al. 2022), and that they allow easier interpretation, more focus is currently put on supervised Machine Learning models like generalized linear models (GLM), support vector machines (SVM) or random forests algorithms (RF). Still, a correct application and assessment of Machine Learning models is key to reproducible and interpretable research results.

Now, with compositional data and log-ratio transformations, a new step stone should be introduced to Machine Learning. Some problems become apparent when combining microbiome data, Machine Learning, and log-ratio transformations. Firstly, it is not known how relevant the introduced properties log-ratio transformations are for Machine Learning purposes. It seems logical that scale invariance, permutation and perturbation invariance should be of importance, but considering that Machine Learning can theoretically make use of the whole output from a sequencing machine, it is in question if for example sub-compositional coherence is relevant and how this impacts the choice of log-ratio transformations. As an additional problem, microbiome sequencing data is usually very sparse, which means it contains a lot of zeroes and therefore it is left-skewed in its distribution (Gloor et al. 2017).

The zero values are especially problematic in the context of log-ratio transformation. The problem is two-fold. First, the logarithm of zero for any type of log is undefined. Secondly, as also ratios are used, placing a zero in the denominator also leads to errors. Simply removing all features containing zeros, as Aitchison suggested (Aitchison 1982) is not practical, especially in a biological context, as it is quite possible to have features with zero values that have biological relevance, for example when comparing gene expressions between samples.

For Machine Learning, zeros itself are not that problematic, still the sparse nature of microbiome data affects machine learning models on several technical levels: the most problematic is the increasing complexity of models, as it needs to fit more coefficients or have greater depth to account for all features. It has also been shown that too many features produce noise in the training data which can easily lead to overfitting and negatively impacts the predictive power. Similarly, a machine learning model can also not differentiate between biological or technical zeros and therefore may underestimate the importance of denser features.

Studies in the last years contributed to streamlining Machine Learning pipelines and establishing a good-scientific-practice in order to ensure reproducibility and usability, also specifically for microbiome data (Jasner et al. 2021). Additionally, studies showed how to reduce bias and error by recommending specific cross-validation procedures (Tsamardinos et al. 2015) and the influence of sample size has been tried to estimate for various Machine Learning models (Beleites et al. 2013; Bailly et al. 2022; van der Ploeg et al. 2014).

To this day, it is still unclear if and how log-ratio transformation impact the performance in a prediction task. In general, log-ratio transformations seem to outperform raw proportions for classification tasks, but it is not clear how log-ratio transformations relate to the changes in predictive performance and if other factors in a Machine Learning pipeline are affected by log-ratio transformations.

A solution that tries to solve some of these problems is to tailor algorithms to the specifications of microbiome data. One such solution is *CoDaCoRe*. Published by Gordon-Rodriguez et al in 2021 in the paper “Learning Sparse Log-Ratios for High-Throughput Sequencing Data”, *CoDaCoRe* is a novel learning algorithm for finding balances (**Co**mpositional **Da**ta via **Co**ntinuous **Re**laxations). Balances are defined as the log-ratios between geometric means of two or more features of the input variables. Translated, *CoDaCoRe* finds ratios between groups of features that are explanatory for the given classification task. Such ratios are commonly used as biomarkers of gut health e.g., the Firmicutes-to-Bacteroidetes ratio (Crovesy et al., 2020; Magne et al.; 2020).

As *CoDaCoRe* is able to combine more than two features in its log-ratios, it leads to a richer set of features and therefore more flexible models, compared to e.g., ALR, which only uses one reference. Finding balances is usually very computationally taxing, however, in *CoDaCoRe* Gordon-Rodrigues et al. use a deep learning technology called “continuous relaxation” and only approximate the optimization problem, which has the advantage of greatly reducing the runtime.

Thus, *CoDaCoRe* identifies a sequence of balances, in decreasing order of importance, each of which is sparse and interpretable. This makes it a promising algorithm that is created to also work efficiently on big data sets with a lot of features and could be a good alternative to counter the problems that microbiome data brings into Machine Learning and log-ratio transformation. In their paper, the authors compare *CoDaCoRe* against several machine learning models (Lasso, RF and XGBoost) and show that their algorithm does not sacrifice interpretability nor predictive accuracy.

5 IMPLEMENTATION

Recent years made it clear that Machine Learning is a tool that should be available to all scientists, but comes with high complexity and its own pitfalls, even without the addition of mathematical characteristics of compositional data and log-ratio transformations. Solutions like *CoDaCoRe* have been introduced by the scientific community, which showed that tailored algorithms could be an alternative to standard Machine Learning models (Gordon-Rodriguez et al. 2021). The goal of this project focuses on collecting insights on the performances of Machine Learning models in direct comparison with *CoDaCoRe*, but also assessing the influence of log-ratio transformations on Machine Learning performances in more detail. Additionally, some practical guidelines will be created on how to effectively combine Machine Learning and compositional data.

As mentioned in the introduction, two log-ratio transformations seem to be the most promising, in terms of applicability and interpretability: ALR and CLR. Additionally, it has been decided to compare ALR transformation in three ways: (i) a random reference will be picked as denominator, to assess the average performance of machine learning models for ALR (random ALR), (ii) find the most optimal denominator (optimal ALR) and (iii) worst ALR denominator (worst ALR).

According to his criteria in Greenacre et al. (2021), the reference component should have a high Procrustes coefficient to maximize the similarity to the original geometry and a low variance. Therefore, for optimal ALR a feature has been chosen that has a high Procrustes coefficient and a low variance and for worst ALR it is the other way around.

A standard normalization for raw relative abundances is total sum scaling transformation (TSS). In theory, it removes technical bias related to different sequencing depth by dividing each feature count with the total library size. As mentioned in the introduction, Aitchison criticized these normalization techniques very early on and therefore its impact on Machine Learning performances will be directly compared to chosen log-ratio transformations. Finally, all Machine Learning performances are compared to *CoDaCoRe* in the following conceptual framework:

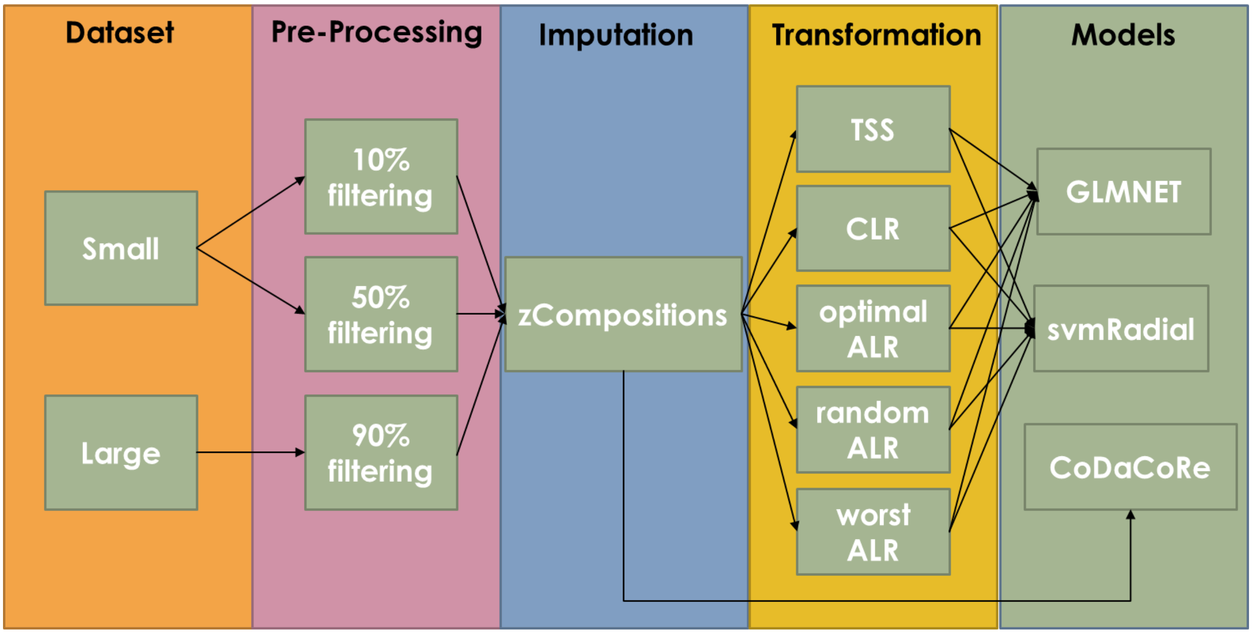

Figure 5: Pipeline to compare model performances

The figure shows the proposed pipeline for effective comparison of machine learning performances. Data sets used are chosen by their characteristics large/small, high/low correlations and continuous/discrete variables. Afterwards, data sets will be filtered by abundances and imputed. Microbiome-native methods will be employed and compared to the data being log-transformed and used in machine learning models.

The general pipeline will be constructed of the following building blocks: Pre-processing, Imputation, Transformation, and Machine Learning Models/Microbiome Approaches. Usually, model performances are averaged after several repeats with random training and test splits. However, it was decided here to use the same training/test split for all transformations to directly compare the difference.

As representation for Machine Learning models, regularized Generalized Linear Model (*glmnet*) and Support Vector Machine (*svmRadial*) are used. *Glmnet* represents a common linear Machine Learning model and *svmRadial* a non-linear one, as several papers already showed different performances for microbiome data depending on the choice of model (Zhang and Shi 2019; Kubinski et al. 2022; van der Ploeg et al. 2014).

For *CoDaCoRe*, it was decided to train two models per run, one with lambda = 0 and the other with lambda = 1. In *CoDaCoRe* the parameter lambda controls the regularization strength of the model. Lambda = 1 applies the 1-standard-error rule in the discretization step of the log-ratio. This is typically a good choice, leading to models that are both sparse and predictive. Lambda = 0 corresponds to a 0-standard-error rule, in other words choosing the log-ratio that minimizes cross-validation score (Gordon-Rodriguez et al. 2021). Using both models allow for a direct comparison in how *CoDaCoRe* handles different types of microbiome data sets, that are typically very sparse. Additionally, *CoDaCoRe* allows in prediction tasks to include all found log-ratios or just the most descriptive one. It was decided to use both to see how influential one found log-ratio can be for the chosen data sets or if it is better to use several log-ratios.

Performance assessment scores are necessary to compare all models directly. For discrete parameters (e.g., two binary classes like “healthy” and “non-healthy”) AUROC (area under the receiver operating characteristic) will be used to assess model performances. AUROC indicates the model’s ability to discriminate between cases (positive examples) and non-cases (negative examples). An AUROC above 0.8 generally means excellent discriminatory ability for the model, whereas an AUROC of 0.5 corresponds to a coin flip (i.e., a useless model).

For continuous parameters (e.g., BMI, which can have any real number) a root mean squared error (RMSE) is used for performance evaluation. It is the standard deviation of the residuals. Residuals are a measure of how far from the regression line data points are distributed and an RMSE consequently measures how spread out these residuals are. A value of 0 would indicate a perfect fit to the data. Contrary to the AUROC, the lower the RMSE, the better the model.

As several authors pointed out Machine Learning performance is influenced by data size (Quinn and Erb 2020; Gloor et al. 2017; van der Ploeg et al. 2014; Bailly et al. 2022). Therefore, three data sets were chosen accordingly to include direct comparison of performances of small and large data sets, as well as high and low known correlations between microbiome and host. We treat these characteristics as part of the analysis to compare model performances. There is no explicit definition for small datasets in the literature. However, several studies classified data sets ranging from 18 to 1030 samples as small data sets (Althnian et al. 2021) and this guideline is used here, too.

This leads to two small (CRC and PCOS) and one large data set (EstMB) for this project. The CRC data set was first used and described by Wirbel et. Al (2019) in their meta-analysis for colorectal cancer. This data set contains 7727 features with 695 samples. It shows clear associations between gut microbiota and colorectal cancer and is therefore helpful to show the behavior of transformations and machine learning algorithms on small but highly specific data sets. The second data set is the Polycystic Ovary Syndrome (PCOS) data set described by Lüll et al. (2021). It observed 312 individuals, with two-thirds of them being healthy, and 72738 features. It is a valuable addition as it is a small data set that shows weak associations between the disease and microbiome structure (Lüll et al. 2021). Lastly, is the Estonian Biobank microbiome cohort (EstMB). This data set includes 2509 individuals with several phenotypical markers collected over time and 17180 features overall, which classifies it as a big data set and it contains two phenotypes that are of value for this project: Hypertensive heart disease (HHD) which has a lower to moderate correlation with gut microbiota and Diabetes Type 2

(DT2) which is known to show a higher correlation to gut microbiome compositions (Gacesa et al. 2022).

To assess if the feature size is an important parameter in Machine Learning performances, it has been decided to apply an abundance filter on all data sets. This has the advantage of removing the majority of zeros and solves therefore some of the issues described above.

6 METHODOLOGY

For data analysis and model pipelines, the script language R (v4.1.3) in combination with RStudio (v2022.02.1+461) has been used. For data cleaning and filtering the main library is “tidyverse” (1.3.1). Imputation was conducted with “zCompositions” (1.4.0.1), and transformations were done with “easyCODA” (0.34.3). Models were constructed with “mikropml” (1.2.2), “tidymodels” (0.2.0) and “codacore” (0.0.3).

Additionally, scripts were created for convenience purposes. All codes can be found on Github JenniferNeumaier/ml_coda.

6.1 PRE-PROCESSING

First, all data sets were cleaned in order to remove NAs in predictor columns or patients that have no sequencing data. In EstMB data set, 21 rows removed in metadata due to NA and 21 patients respectively cut out of abundance table. This leads to N = 2485 final sample-size. In CRC, 128 rows were removed due to NA in feature “BMI”, leading to 567 samples overall. In PCOS, 6 rows were removed in the abundance table because no matching patient has been found in metadata, reducing the number of samples to 304.

As discussed, the sparse and high-dimensional nature of microbiome data leads to several problems. Therefore, filters were applied to all three data sets. In this benchmarking project, taxa with ≤10% prevalence in the datasets will be discarded. Additionally, a filter of ≤50% abundance prevalence in samples will be applied, as well as a mean relative abundance filter for 0.001. For 10% prevalence filters CRC keeps 650 features and PCOS 1154 features. Respectively, for 50% prevalence filters CRC keeps 189 features and PCOS 120 features. For EstMB data, 90% prevalence filter was used, as the data was otherwise not practically usable without heavy computational power. Even with 90% filtering, EstMB keeps 3062 features and with 95% filtering 2589 features. If not otherwise stated, all tests use the 10% prevalence filtered data sets for PCOS and CRC and 90% filtered data sets for EstMB.

6.2 IMPUTATION

One of the first steps after filtering and before log-ratio transformation is zero-imputation. Introduced by (Palarea-Albaladejo and Martín-Fernández 2015) is the library “zCompositions”. All three data sets were imputed with Geometric Bayesian Multiplicative (GBM) and output form “p-counts”.

6.3 TRANSFORMATIONS

Greenacre et al. (2021) proposed a computational solution of finding an ALR reference. Taken from his Github (https://github.com/michaelgreenacre/CODAinPractice) is the function FINDALR() that assesses the abundances and variances of features in a data matrix, followed by a Procrustes analysis to assess the similarity to the original geometry. It therefore calculates the variances for all features as well as a Procrustes coefficient. Custom functions were created for all ALR functions that uses this function, as well as ALR() from the library “easyCODA”. Similarly, “easyCODA” also contains the function CLR() to compute the centered log-ratio. For TSS() a custom function has been created.

6.4 MACHINE LEARNING MODELS

The focus will be on two standard Machine Learning models, that are already incorporated in easy-to-use packages in R: generalized linear models (GLMs) and support vector machines (SVM) as non-linear approach. Both are available in the function run_ml() from the package “mikropml” by Topçuoğlu et al. (2020). This package nicely compacts the use of standard machine learning models to a few lines of code and supports the use of GLMs (*glmnet*), as well as SVMs (*svmRadial*).

“Tidymodels” was used split the data into 80% training set and 20% test set. A seed was used to ensure that the data splits were the same for every repeat and following test. The split data has then been fed into run_ml() and codacore(), respectively. Both functions then split the training set into 5 folds and assessed via 10-times repeated cross-validation the best model. Here, run_ml() also tunes hyperparameters for *glmnet* and *svmRadial*, respectively. In *CoDaCoRe* lambda is not tuned, but instead two models are trained, one with lambda = 0 and the other with lambda = 1. The best model then is used for predictions on the test set and its AUROC/RMSE scores saved.

This procedure (without a new data split) is repeated 10 times.

For *svmRadial* the number of repeats had to be lowered, otherwise the training of the model would have taken too long. It has been decided to split the training set in 5 folds and apply a 2-times repeated cross-validation and 5 repeats overall (without a new data split).

7 RESULTS

7.1 INFLUENCE OF TRANSFORMATIONS ON ML PERFORMANCES

The most important research question of this thesis was if and how log-ratio transformations influence Machine Learning performances. The pipeline introduced in section 3 has been applied to all three data sets and Figure 6 shows the culminated results of the presented pipeline and combined in one figure to show the differences of Machine Learning and log-ratio transformations on several data sets directly.

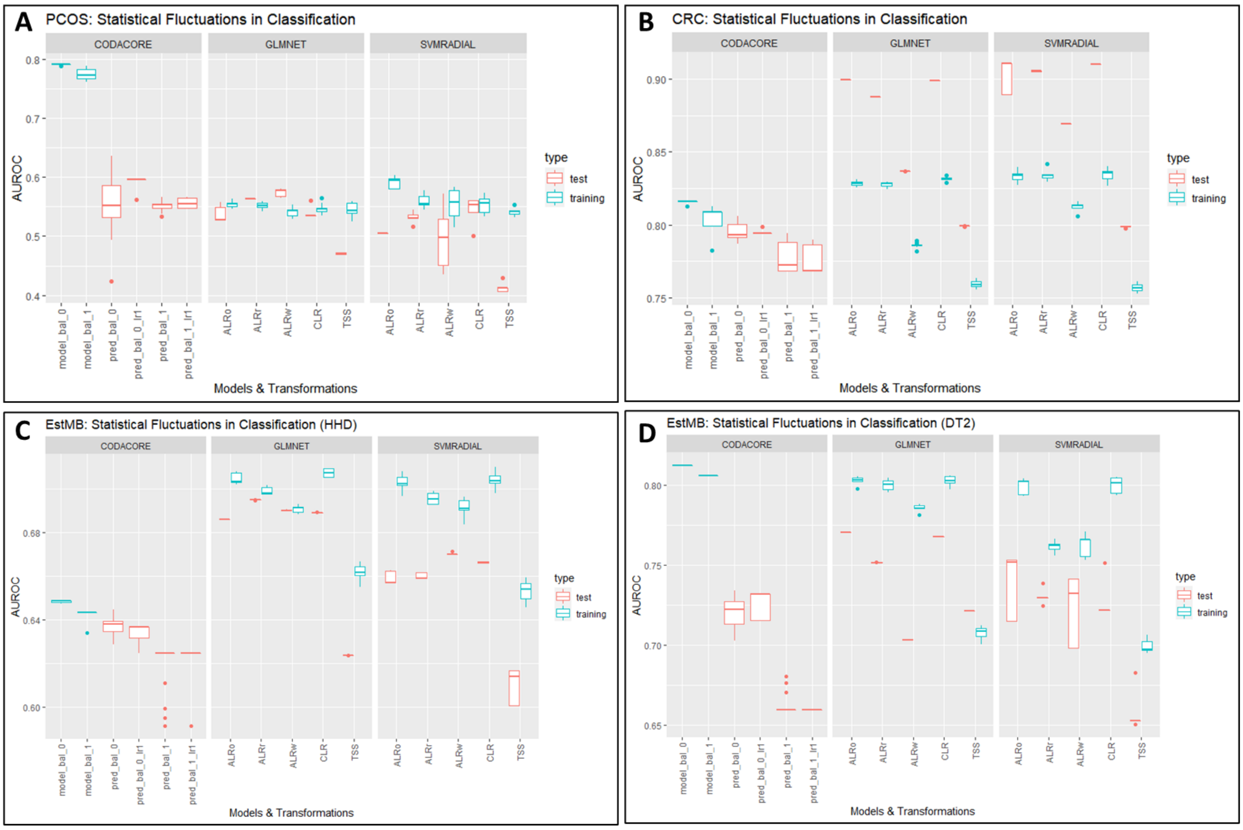

Figure 6: Statistical Fluctuations in Classification

(A) contains data from PCOS data set predicting PCOS phenotype. (B) contains data from CRC data set predicting CRC phenotype. C and D both contain data from EstMB data set, with (C) predicting for hypertensive heart disease (HHD) and D for diabetes type 2 (DT2).

The y-axis contains AUROC scores. The x-axis contains models and transformations, with the most left always being *CoDaCoRe*, the middle portion displaying performances of *glmnet* and the right *svmRadial*. In red test performances are plotted and in blue training performances. model_bal_0 and model_bal_1 correspond to *CoDaCoRe* models with lambda 0 and lambda 1, respectively. Consequently, pred_bal_0 and pred_bal_0_lr1 correspond to predictions with all found log-ratios and only the most descriptive log-ratio for model_bal_0. The same nomenclature applies to pre_bal_1 and pred_bal_1_lr1. ALRo, ALRw and ALRr correspond to the types of ALR transformation applied in this pipeline (optimal, worst and random).

When comparing Machine Learning models and algorithms, several points stand out. As expected for PCOS (Fig.6A), the AUROC scores for test and training performances vary from

0.5 to 0.6 and correspond to the low association between microbiome and PCOS. In the CRC data set, all Machine Learning models severely underfit the data, which usually indicates a bias in the data. A possible reason could be a “leakiness” in the pipeline. It is considered good practice to not conduct pre-processing on the whole data set (Whalen et al. 2022). Although imputation and transformation are handled independently from pre-processing in this pipeline, both can be considered part of pre-processing. Indeed, section 5.2 shows that splitting the data set before imputation and transformation solves the underfitting in CRC. Additionally, linear models (*glmnet*) seem to perform better than non-linear (*svmRadial*), although that difference is smaller in high-association data sets like CRC and EstMB.

CoDaCoRe severely overfits the data set with the test performances corresponding to the performances of both *glmnet* and *svmRadial*. It is surprising, that *CoDaCoRe* shows lower test and training performances, comparable to ALRw-transformed or TSS-normalized data (Fig.6C). In general, its results are very inconclusive throughout different types of data. In the PCOS data set (Fig.6A) it visibly overfits the data, although its test performances are comparable to *glmnet* and *svmRadial*. In high correlation data sets, independent of sample size, it does not manage to perform better than Machine Learning models, although performances are still considered good to excellent with an AUROC of 0.8 (Fig.6B and 6D).

Applying log-ratio transformations has several effects on Machine Learning performances: all models in Fig.6B show good to excellent performances, with ALRo-transformed and CLR-transformed data improving model performances even more. Machine Learning models with ALRr-transformed data shows surprisingly good results, which is repeated in Fig.6C and D for the EstMB data set. Although log-ratio transformation not necessarily improve the predictive performance in low correlation data set (Fig.6A), Machine Learning models with TSS-normalized data still show the lowest performances in all three data sets, independent of correlation or sample size. This trend is also visible in regression settings for all three data sets (see Supplementary Fig.10).

Still, the trend depicted in CRC with ALR- and CLR-transformed data producing best Machine Learning performances, is also found in EstMB data sets, for both moderate correlation (HHD) and high correlation (DT2) between microbiome and disease, and to a small degree in PCOS. Considering the effect of transformations even in PCOS, it seems to be a clear indication, that log-ratio transformations can indeed affect the performance in machine learning models. Additionally, *CoDaCoRe* only performs as good as log-ratio transformed data, but never better.

7.2 DATA LEAKAGE

As mentioned in the section above, data leakage is an easy to overlook problem and already lead to performance biases in the section before, seemingly more in smaller data sets. Information leakage occurs when information from the test set is used to pre-process the training set. This is for example happening when the imputation algorithm uses the whole data set, which includes the test set, to replace zeroes (Whalen et al. 2022). To assess the impact of data leakage when adding log-ratio transformations, this section directly compares the model performances of *glmnet* and *CoDaCoRe* of a leaky and non-leaky pipeline.

In theory, the chosen log-ratio transformations ALR and CLR should not lead to any kind of data leakage. CLR transformations are conducted row-wise and therefore do not use information between samples. For ALR, if the analysis for the best reference is conducted on the training set and applied on the test set, then data leakage should also not occur. The same applies if a random reference is chosen. Thus, a closer look should be given to imputation procedures in terms of data leakage.

In the non-leaky procedure, processed data is split into train and test set at the start and imputation and transformations are applied separately on the split data. In the case of ALR, the reference was found on the training set and directly applied to the test set. Finally, train and test set are fed into *glmnet* model. The non-leaky procedure for *CoDaCoRe* works similarly but the transformation step is skipped.

In the leaky procedure, imputation and transformations are conducted on the whole data set and afterwards the data is split into train and test set and fed into the *glmnet* model. Again, for *CoDaCoRe* the transformation step is skipped.

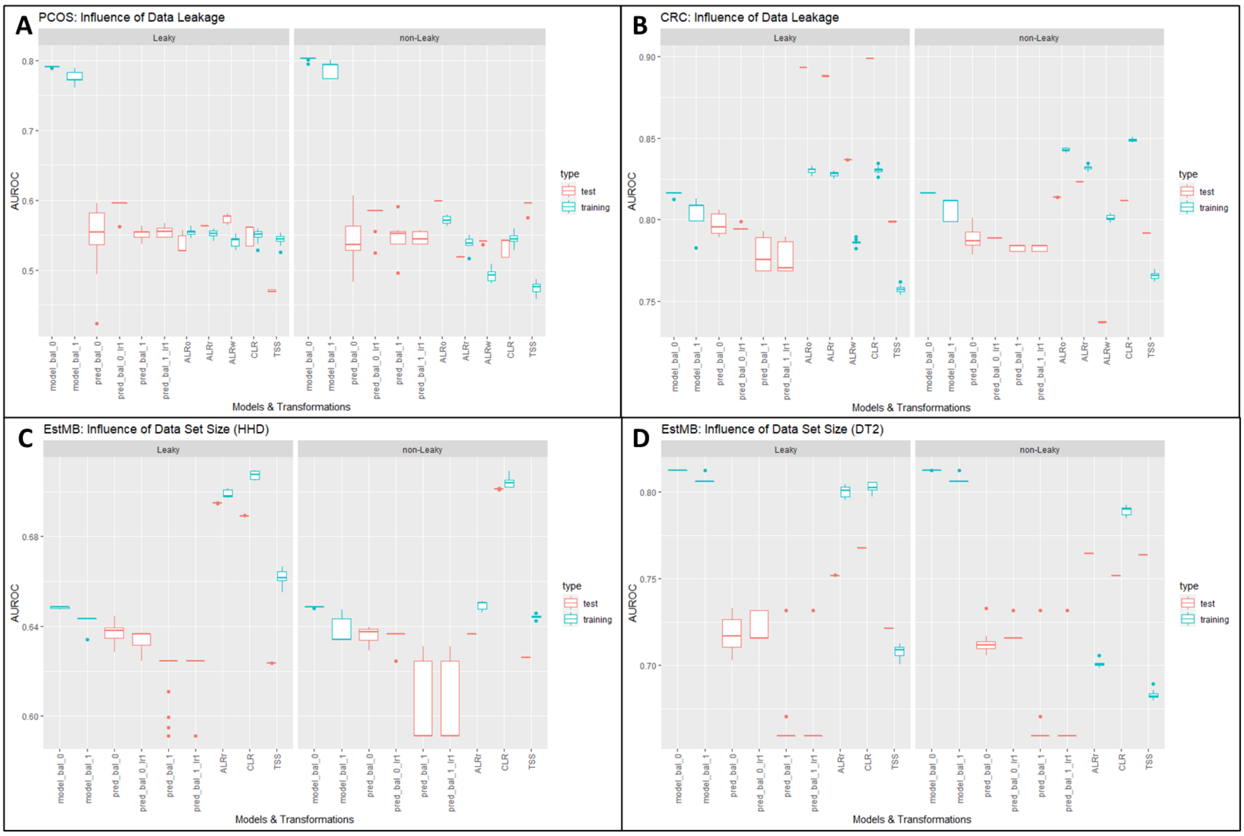

Figure 7: Influence of leaky and non-leaky pipelines on log-ratio transformation

(A) contains data from PCOS data set predicting PCOS phenotype. (B) contains data from CRC data set predicting CRC phenotype. C and D both contain data from EstMB data set, with (C) predicting for hypertensive heart disease (HHD) and (D) for diabetes type 2 (DT2).

The y-axis contains AUROC scores. The x-axis contains models and transformations, with the most left always being *CoDaCoRe,* followed by *glmnet* results. For EstMB ALRo and ALRw have been skipped because it was too computationally taxing to repeat a Procrustes Analysis. In red test performances are plotted and in blue training performances. model_bal_0 and model_bal_1 correspond to *CoDaCoRe* models with lambda 0 and lambda 1, respectively. Consequently, pred_bal_0 and pred_bal_0_lr1 correspond to predictions with all found log-ratios and only the most descriptive log-ratio for model_bal_0. The same nomenclature applies to pre_bal_1 and pred_bal_1_lr1. ALRo, ALRw and ALRr correspond to the types of ALR transformation applied in this pipeline (optimal, worst and random). *Glmnet* has been used to train all models.

The results for the leaky pipelines were not repeated and the description for the results can be read in section 5.1.

It becomes apparent that the results for *CoDaCoRe* do not change significantly if imputation is conducted on the whole data set or not. In CRC and EstMB (Fig.7B and 7D), imputation on the whole data set potentially leads to higher variances in test performances compared to splitting the data before imputation.

The results for the standard Machine Learning models are a bit more mixed. The results in Fig.7B show that the underfitting in the CRC data set was most likely due to biases in the pipeline. As soon as a non-leaky was applied, the Machine Learning models are not underfitting the data anymore. The training performances for CRC are very similar to the leaky pipeline, but the test performances have been adjusted. A contrary picture is shown for TSS-normalized data in PCOS and EstMB DT2 (Fig.7A and 7D). There, the model is now heavily underfitting the data, whereas no significant changes are seen in Fig.7B (CRC) and 7C (EstMB HHD) for TSS.

Some trends can be observed for all three data sets: CLR-transformed data leads to very stable Machine Learning performances (sans the underfitting in CRC), which is especially visible in EstMB performances. Applying ALR or TSS on data impacts Machine Learning performances greatly by the changed pipeline. In all data sets Machine Learning performances with TSS-normalized data increases the difference between its test and training performance, independent if it has under- or overfit the data. In Fig.7A and 7B Machine Learning performances with ALRo-transformed data marginally underfits the data but otherwise shows the highest performances out of all transformations.

Even though a leaky pipeline has been applied in the section before, the trends applied by log-ratio transformations still holds. In both pipelines, log-ratio transformations outperform standard normalization techniques and *CoDaCoRe* is not outperforming standard Machine Learning models. Nevertheless, in all further tests, the non-leaky pipeline will be applied.

7.3 INFLUENCE OF TEST SET METHODOLOGY

The pipeline used to compare all data sets and their performances includes a standard data set split of 80/20, meaning 80% of data goes into the training set, 20% of data into the test set. However, the CRC data set has been designed with holdout sets in mind and the original paper uses this technique (Wirbel et al. 2019). In this paper they called it leave-one-study-out (LOSO) validation, where data from one study was set aside as test set, while the data from the remaining 4 studies was pooled as a training set (Wirbel et al. 2019).

The study design gives a good opportunity to see if the influence of log-ratio transformations is impacted by the choice of test set methodology. The following figure compares *glmnet* and *CoDaCoRe* performances with a standard 80/20 or Holdout set.

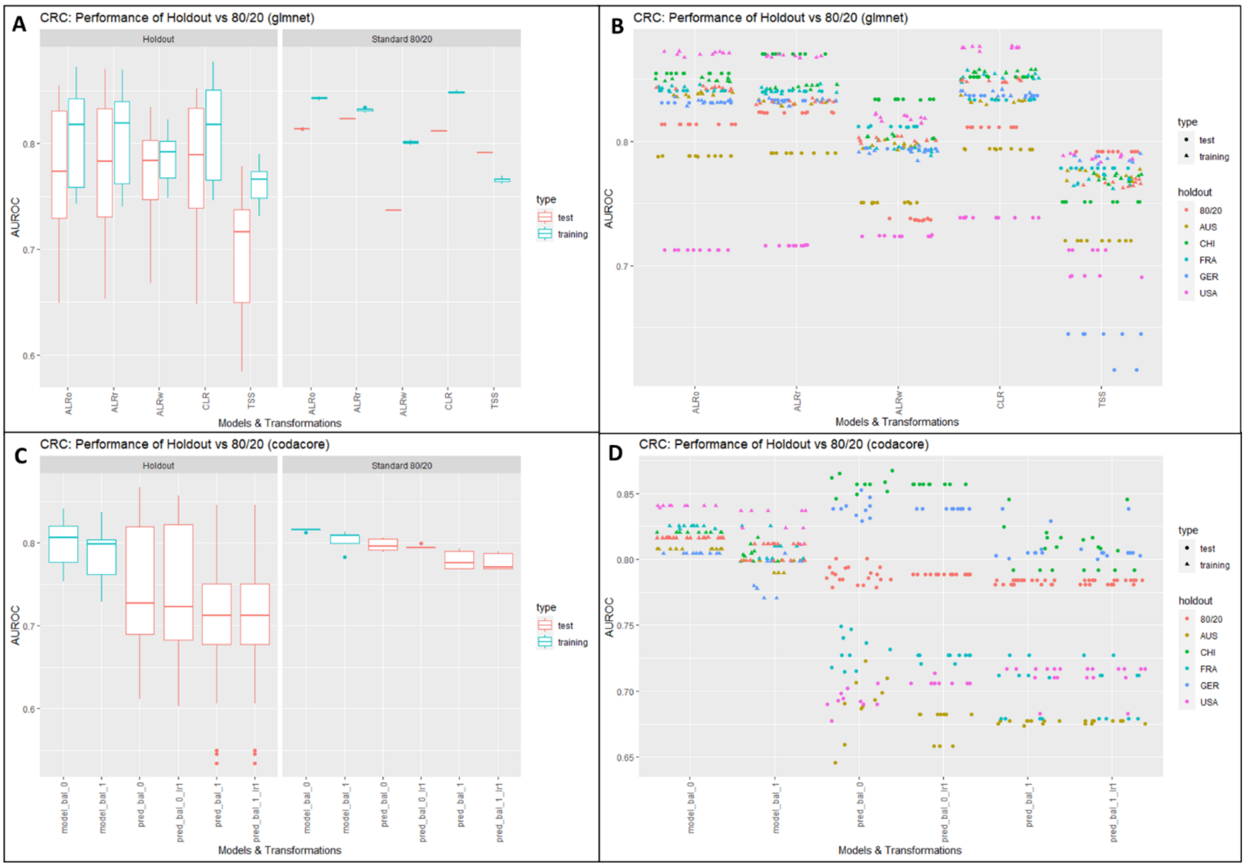

Figure 8: Behavior of log-ratio transformations with test set methodology

(A) contains data from CRC data set predicting CRC phenotype and (B) plotting all test and training performances in detail. The same applies in (C) and (D) for CRC *CoDaCoRe* performances.

The y-axis contains AUROC scores. The x-axis contains models and transformations, with A and B belonging to glmnet models and C and D belonging to *CoDaCoRe*. In red test performances are plotted and in blue training performances. model_bal_0 and model_bal_1 correspond to *CoDaCoRe* models with lambda 0 and lambda 1, respectively.

Consequently, pred_bal_0 and pred_bal_0_lr1 correspond to predictions with all found log-ratios and only the most descriptive log-ratio for model_bal_0. The same nomenclature applies to pre_bal_1 and pred_bal_1_lr1. ALRo, ALRw and ALRr correspond to the types of ALR transformation applied in this pipeline (optimal, worst and random). *Glmnet* has been used to train all models.

Fig.8A and 8B show the difference of test performances as boxplots for different transformations of the same data set and for *CoDaCoRe*, respectively. All five holdout sets have been used separately as test set once and the results pooled. It can be seen clearly that for all transformations, the holdout sets show a lot more variance in the test set performances, compared to the standard 80/20 split, which shows almost no variance at all and arguably higher test performances. This is also visible for *CoDaCoRe* performances.

In Fig.8B and 8D it becomes clear why the test performances of the holdout sets have such a high variance. For all transformations and holdout sets, the test performance is quite stable. Marked in pink dots is the test set performance when USA has been chosen as holdout set. For all ten repeats of the model, the test performance is stable around 0.72. However, choosing CHI as the holdout set produces test set performances of 0.85. This shows that the high variances in Fig.8A are not due to statistical fluctuations, but the choice of holdout set. Splitting the data set in an 80/20 manner leads to a performance somewhere in the middle. This is also true when using *CoDaCoRe,* although the test set performances fluctuate more strongly.

Conclusively, applying log-ratio transformations impacts Machine Learning performances similarly to results in section 5.1 and 5.2. From these results it becomes obvious that log-ratio transformations behave consistently independent on the choice of test set. The performances of Machine Learning models with ALRo-, ALRr- and CLR-transformed are higher than with TSS-normalized and ALRw-transformed data, even when using holdout sets. *CoDaCoRe* doesn't change its performance.

In that sense, Machine Learning performances in section 5.1 for CRC could be higher, when choosing a specific country as test set. It can be argued however, that this introduces a heavy bias, as the difference between some test set performances are very distinct. The results show that choosing a standard 80/20 split averages out these differences and is perhaps more beneficial if data is more unknown and less planned as the CRC data set.

7.4 INFLUENCE OF NUMBER OF FEATURES

As mentioned in the introduction, the high dimensionality and sparse nature of microbiome data is problematic for both Machine Learning models and log-ratio transformations. As a common technique to reduce the runtime for pre-processing and zeroes contained in the microbiome data, a prevalence filter is applied, or the features are selected via ordination techniques (Jasner et al. 2021; Rivera-Pinto et al. 2018).

Applying prevalence filters for all three data sets leads to the following losses in the number of features: 10% prevalence filters leave 650 features for CRC and 1154 features for PCOS. Respectively, for 50% prevalence filters CRC keeps 189 features and PCOS 120 features. Applying a 90% prevalence filter on EstMB data set leaves 3062 features and applying 95% filtering leaves 2589 features. Consequently, 10%/50% prevalence filters lead to a loss of over 90%/97% of features for CRC and over 98%/99% for PCOS, respectively. In EstMB the loss is only 83%85% for 90%/95% filtering, respectively.

It would be of great advantage if the number of features could be reduced even further, with Machine Learning performances staying stable. Therefore, the following test will assess how Machine Learning models perform if a stricter prevalence filter is applied. Additionally, *CoDaCoRe* performances will be again compared to *glmnet*.

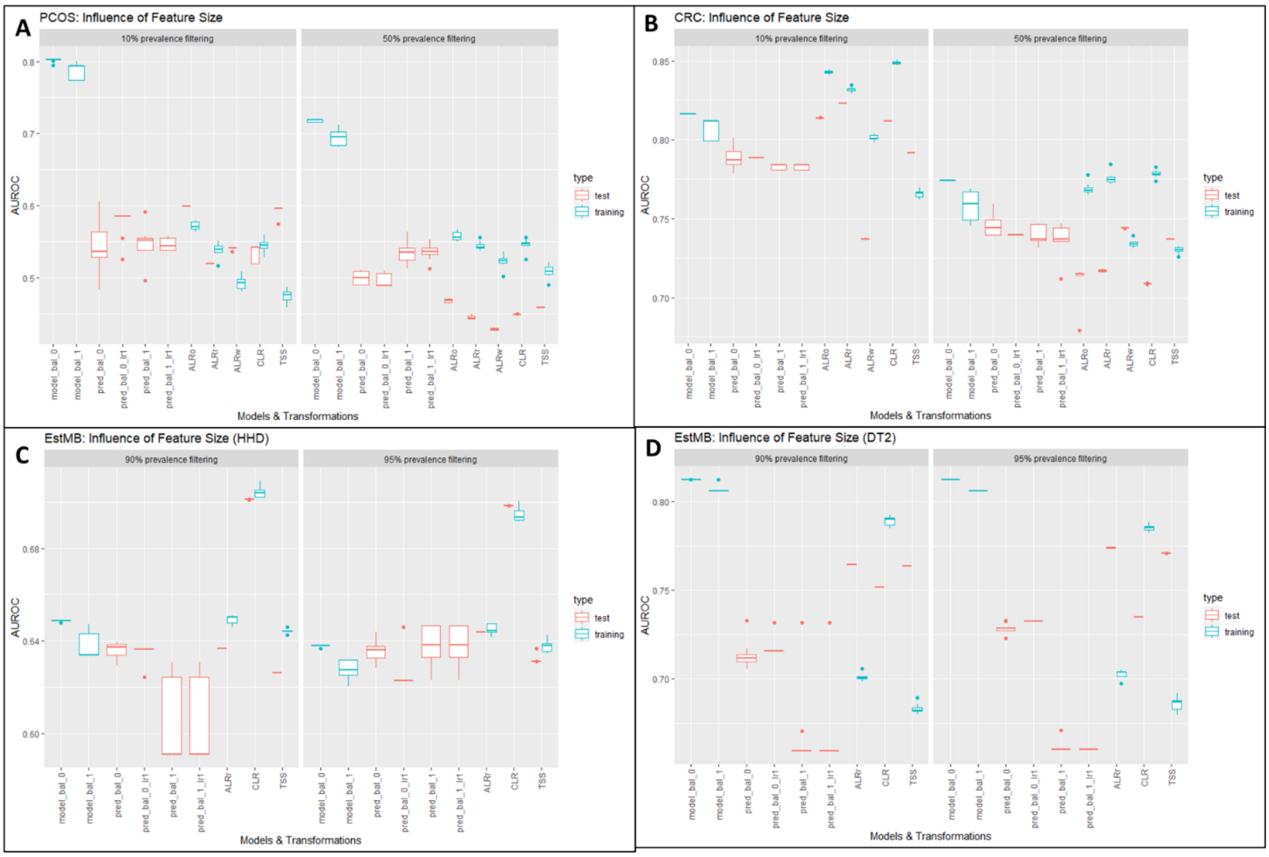

Figure 9: Influence of Feature size

(A) contains data from PCOS data set predicting PCOS phenotype. (B) contains data from CRC data set predicting CRC phenotype. (C) and (D) both contain data from EstMB data set, with C predicting for hypertensive heart disease (HHD) and D for diabetes type 2 (DT2). For CRC and PCOS Machine Learning performances with 10% prevalence filters were compared to 50% prevalence filters. For EstMB 90% prevalence filters were compared to 95% prevalence filters.

The x-axis contains models and transformations, with the most left always being *CoDaCoRe*, the middle portion displaying performances of glmnet and the right svmRadial. In red test performances are plotted and in blue training performances. model_bal_0 and model_bal_1 correspond to *CoDaCoRe* models with lambda 0 and lambda 1, respectively. Consequently, pred_bal_0 and pred_bal_0_lr1 correspond to predictions with all found log-ratios and only the most descriptive log-ratio for model_bal_0. The same nomenclature applies to pre_bal_1 and pred_bal_1_lr1. ALRo, ALRw and ALRr correspond to the types of ALR transformation applied in this pipeline (optimal, worst and random). *Glmnet* has been used for all models.

The results for 10% prevalence filters were not repeated and the description for the results can be read in section 5.2, as the non-leaky pipeline has been applied.

Reducing the number of features has visible impacts on smaller data sets (Fig.9A and B). For PCOS, *glmnet* is overfitting the data more pronounced compared to 10% filtered data sets and in CRC, the decrease in performance is clearly visible (Fig.9B), accompanied by heavier overfitting. The impact on bigger data sets is smaller, however. In Fig.9C a 95% prevalence filter leads to less distance between test and training performances compared to 90% filtering, whereas these distances increase slightly in Fig.9D. In general, though the performances are very similar.

Again, although the performances in CRC (Fig.9B) are decreasing, the impact of log-ratio transformations are still visible. Machine Learning performances of ALRw-transformed and TSS-normalized still perform worse compared to ALRo, ALRr and CLR. Especially the latter is also very stable in EstMB data sets. The performances of *CoDaCoRe* are not heavily affected, only in CRC do they decrease with a decrease in features.

8 DISCUSSION

The goal of this thesis was to assess the influence of log-ratio transformations on Machine Learning performances, specifically in the context of microbiome data. Additionally, it has been tried to assess if some blocks in the Machine Learning pipeline are more affected by the addition of log-ratio transformation.

8.1 MACHINE LEARNING

Classification in Machine Leaning is a challenging task by itself, and it becomes more challenging when dealing with small datasets, as the limited size of training data can lead to unreliable models (Althnian et al. 2021). This is important in biological contexts, as sample sizes are usually limited, and it should be accounted for when Machine Learning models are applied and evaluated.

The CRC data set shows that high sample-sizes are not strictly necessary in highly associates data sets, which is supported by several other studies in recent years (Althnian et al. 2021; van der Ploeg et al. 2014; Bailly et al. 2022). Especially for logistic regression models (like *glmnet*), smaller data set sizes are sufficient to achieve quick and good performances (Bailly et al. 2022). Experts are in agreement that an overall performance of classifiers depends how well the data set represents the original distribution rather than its size (Althnian et al. 2021) and that the most robust classifier is not necessarily the best one (Althnian et al. 2021; Bailly et al. 2022). However, it can be approximated that a number between 75–100 samples will usually be needed to test a good but not perfect classifier (Beleites et al. 2013). These results are supported by comparing PCOS and CRC Machine Learning model performances. The PCOS test set contains only 60 samples but has low variances in training and test set performances, which indicates a low uncertainty.

For non-linear models, it has been shown that SVM shows instabilities in smaller data sets and mentions that SVM may need over 10 times as many features per phenotype to achieve a stable AUC compared to linear models (van der Ploeg et al. 2014). Therefore, they could be better equipped to deal with the high dimensionality and sparse nature of microbiome data. When comparing several Machine Learning models, several studies showed that non-linear models usually perform better than linear ones (Kubinski et al. 2022; Zhang and Shi 2019). However, this trend is not depicted in the results here. *SvmRadial* shows similar performances compared to *glmnet* in small data sets (section 5.1, PCOS and CRC), but shows reduced performances in the large data set. However, further tests and comparisons with e.g., XGBoost are necessary as non-linear algorithms may need more time and repeats to train models, which could correct the here depicted results.

Comparing CRC and EstMB (DT2) results it is clear that excellent Machine Learning performances in high-correlation data sets can be achieved independent of data size. But these results do not show if the AUROC scores could be higher with an even bigger sample size. An additional test with the EstMB data set could give more insight, by e.g., creating several differently sized data sets out of it and comparing the performances. If the performances increase with the sample size it could be an indicator that also log-ratio transformations are directly affected by sample sizes. Especially in the case of ALR this dependency should be investigated further. Similar to the problem of data leakage, all procedures that are conducted column-wise are affected by sample size and the calculation of variances in the Procrustes analysis is no exception.

8.2 LOG-RATIO TRANSFORMATIONS OR CODACORE?

Overall, it can be stated that log-ratio transformation indeed increases Machine Learning performances, compared to standard normalization. These results are backed by several other studies, which showed similar results and recommended the use of log- or log-ratio transformation (Quinn and Erb 2020; Jasner et al. 2021; Kubinski et al. 2022). Furthermore, it can be stated that any choice of log-ratio transformation increases Machine Learning performances. Although only CLR and ALR have been tested in this study, recent studies have shown that ILR also seems to improve performances (Kubinski et al. 2022). This effect is less pronounced in data sets that show weak associations (see section 5.1) as the increase in performances is only marginal. This indicates that log-ratio transformations do not magically conjure more associations.

Furthermore, in high-correlation data the choice of transformation seems almost detrimental as CLR and optimal ALR show similar good performances. Choosing a random feature for ALR shows surprisingly good performances, which would support recent claims that strict-isometry may not be necessary for machine learning purposes (Greenacre et al. 2022). Therefore, choosing a random ALR feature could be very beneficial as it is not as computationally taxing as calculating the optimal ALR, especially in bigger data sets. It should be noted however, that random reference features could lead to performances as indicated by ALR worst, with a chance of 1/number of features. This probability becomes more redundant the more features are used and as an additional factor, the impact of a bad reference feature on the performance seems to decrease with sample size. In CRC, ALR worst shows performances close to TSS, whereas in EstMB data sets the performance is closer to the other ALR performances, independent of phenotype correlation. This would support the hypothesis by Greenacre et al. (2022) that deviation from isometry and sub-compositional coherence might become more diluted in higher-dimensional data and is not as necessary anymore.

The fact that log-ratio transformation all increase Machine Learning performances has the advantage that the researcher may choose an appropriate log-ratio transformation according to the research question. Furthermore, even in pooled data sets and additional steps like batch corrections log-ratio transformed data performs better than classically normalized one, which suggests that log-ratio transformations could also work in highly individualized pipelines (Kubinski et al. 2022).

One aspect of log-ratio transformation that has not been touched in this project is that of interpretability. Although we successfully map compositional data from the Simplex Space into Euclidean Space with log-ratio transformation and can apply any commonly known downstream analysis from this point on, it comes with a slight cost of different interpretability. ALR is the most straight-forward as the change in one feature can be interpreted *in reference* to the chosen denominator, whereas CLR has to be interpreted that any change in a feature is in reference to the geometric mean of the whole composition.

Thus, if one is interested in using Machine Learning models for feature selection or finding predictive features, choosing ALR could make the final interpretation easier than CLR. On the other hand, if the interest lies in stable prediction and classification models, where interpretation beyond AUROC scores is not necessary, CLR could be the best choice, with perhaps a log to the base of 2 (Coenders and Pawlowsky-Glahn 2020).

In terms of interpretability, *CoDaCoRe* could also be advantageous, as it finds balances between features and automatically selects descriptive features. This creates an interpretation similar to ALRs, but with less predictors compared to *glmnet* models. As a qualitative analysis (see Supplementary Fig.11B) shows, in both 10% and 50% filtering for CRC data sets *CoDaCoRe* finds very similar and overlapping log-ratios, respectively. This suggests that the algorithm for *CoDaCoRe* is indeed quite stable, which would make it more ideal for finding biomarkers or descriptive features for a specific phenotype.

Although *CoDaCoRe* does not outperform Machine Learning models with log-ratio transformed data, it shows good results in high-association data like CRC and EstMB (DT2), with only a few found balances. Even if only the most predictive log-ratio has been used to predict the test set, the performances were never worse than TSS-normalized data. This can be a benefit, as it indicates that *CoDaCoRe* is able to predict well with only a few well selected features, compared to the many predictors used in *glmnet* models.

These results support the notion that in highly associated data, a small number of features is sufficient to produce good models. This claim is also supported by the computer science community. For uncorrelated features, the optimal feature size may be N−1 (with N = sample size) and for highly correlated features $\sqrt{N}$ (Hua et al. 2005). It suggests that more symmetric data sets may be more preferential, which is potentially supported by the results in this thesis. The data sets that perform well (CRC, EstMB) both have more symmetric data sets after filtering than PCOS, with the number of samples comparably close to feature sizes.

Additionally, it has already been shown to improve Machine Learning performances when genus level representation was used and all features belonging to the same genus merged through PCA before Machine Learning (Jasner et al. 2021). Future tests to compare such approaches to amalgamation techniques provided by *CoDaCoRe* could be good choices to move forward.

8.3 KNOWN AND UNKNOWN BIASES IN MACHINE LEARNING

8.3.1 FILTERING

Unfortunately, the method of filtering data (by prevalence or by more deterministic approaches) seem to still affect Machine Learning performances. As can be seen in section 5.4, the performances for all models in small data sets show a decrease in performance after 50% prevalence filtering, compared to 10% prevalence filtering. Large data sets do not seem to be affected. This drop in performance can also be seen in *CoDaCoRe* performances, which is especially surprising, considering its consistency in found log-ratios.

These results suggest that sub-compositional coherence may be a more important factor when combining Machine Learning and log-ratio transformations than indicated by Greenacre et al. (2022). They found very low variability in results when they used several 50% sub-compositions from the same data set. In the case of prevalence filtering, both data sets are sub-compositions of the original, with only keeping features that show up more abundant and should therefore inherit similar results when analyzing ratios.

Still, both *CoDaCoRe* and *glmnet* models show a drop in performance as soon as feature size is reduced. The results hint that feature sizes perhaps need to be assessed as an independent parameter in microbiome data and are a very Machine Learning specific problem. Maybe those parameters should be tested more in detail to achieve stable Machine Learning performances. This problem also ties back into the discussion of zero imputation, which could introduce a lot of bias into the data sets.

Suggestions for a more targeted feature selection besides prevalence filtering before Machine Learning have been proposed, also in compositional data. Most of them target the search for main factors that maximally explain log-ratio variances (Greenacre et al. 2022). However, an argument can be made that such techniques have been used in this project with *CoDaCoRe* and also ALR reference selection. As shown, *CoDaCoRe* does not improve performances although it supposedly found predictive balances and calculating ALR references is indicated to be affected by the sub-composition (see Supplementary Fig.11A).

8.3.2 LEAKINESS AND IMPUTATION

Two factors that potentially introduce biases in Machine Learning models have already been introduced: log-ratio transformations and imputations. In theory, CLR-transformed data should not be affected is it is conducted sample-wise by nature. The results show that Machine Learning performances with CLR-transformed data shows very similar results compared to the leaky-pipeline, which supports the mathematical theory. In the case of ALR, if the reference feature is calculated on the training set, no information leakage should occur, however the results do not show conclusive results in that regard. Independent if a reference feature is calculated or randomly chosen, Machine Learning performances vary heavily between leaky and non-leaky pipelines, sometimes switching from underfitting to overfitting. More repeats for all three data sets with ALR transformations would be helpful in determining if this log-ratio transformation can be as stable as CLR in a non-leaky pipeline, perhaps by investigating the underlying Procrustes analysis a bit more in detail.

CLR results indicate that log-ratio transformations do not explain the significant underfitting in CRC data, which leads imputation seemingly to be the most impacting factor for data leakage in this pipeline. This is supported when comparing Machine Learning performances of data split before imputation and transformation with data split only before imputation, but transformations are conducted on the whole data set (see Supplementary Fig.12). These results suggest that conducting log-ratio transformation on the whole data set was not the main driver of underfitting in the CRC data set.

Incidentally, although EstMB data sets were conducted with the same pipeline as CRC, the models did not overfit the data, especially when comparing both high correlation data sets. An explanation could be that in bigger data sets the bias introduced by assumptions from the imputation algorithm on the whole data set is significantly reduced. In this project, zCompositions (Palarea-Albaladejo and Martín-Fernández 2015), specifically Geometric Bayesian Multiplicative (GBM), which uses the underlying distribution of a feature to infer a zero replacement. This underlying distribution is much more defined with bigger sample sizes and much less uncertainty is contained in inference processes.

The scientific community is consistently discussing over the impact of imputation for any type of data analysis and this discussion transferred to the application of log-ratio transformations (Quinn et al. 2018; Greenacre et al. 2022; Gloor et al. 2017; Talwar et al. 2018). With the R library “zCompositions” a tool has been created to impute zeroes with compositional data in mind. Alternative imputation algorithms exist, that facilitate Autoencoders like in (Talwar et al. 2018) or are designed specifically for sparse microbiome data like mbImpute (Jiang et al. 2021). The goal of mbImpute is to provide an approach that tries to decide between technical and biological zeroes by leveraging additional information from sample covariates and taxon phylogeny.

Considering the complexity of this topic and its relevancy for both log-ratio transformation and Machine Learning, it would be best to circumvent imputations altogether. Recent papers suggested that it could be possible to circumvent the problem of zero replacement altogether by applying Box-Cox power transformation or chi-square distances (Greenacre et al. 2022). Furthermore, data-driven alpha-transformations are currently investigated that enables one to deal with the presence of zeroes in the compositions directly (Clarotto et al. 2022). Further studies would be necessary to assess if there is a “best choice” in how to deal with zeroes in microbiome data, especially in smaller data sets, and also including filtering approaches.

9 CONCLUSION

Regardless of the mentioned problems in the section before some practical key notes on transformations in the context of Machine Learning can be extracted from this thesis: (1) log-ratio transformations are not able to conjure associations when there are none, (2) any log-ratio transformation is better than none.

This comes with several advantages when combining compositional data with Machine Learning:

Log-ratio transformations can be easily integrated into existing pre-processing procedures and if the data set is too small to conduct log-ratio transformations separately on training and test set, choosing ALR or CLR circumvents this problem easily. Otherwise, any type of log-transformation works, as several studies have already shown (Jasner et al. 2021; Greenacre et al. 2022), and the library "easyCODA" makes many log-ratio transformations easily applicable. Alternatively, *CoDaCoRe* offers an easy-to-use solution, especially if one is interested in biomarkers for a specific phenotype. At this point, log-ratio transformed data and standard Machine Learning tools outperform *CoDaCoRe,* however, previous section indicated that the problems of filtering, imputation and leakiness in this thesis was due to the character of Machine Learning and microbiome data and not necessarily introduced by log-ratio transformations or the *CoDaCoRe* algorithm.

As Imputation and Filtering methodology seem to be the most prevalent problems in applying Machine Learning methods on microbiome data, further tests would be necessary to reduce bias and increase performances. Building on top of this project, it should be tested how amalgamation in *CoDaCoRe* and feature selection via PCA or phylogenetic trees impact Machine Learning performances and especially methods that circumvent imputation entirely would be beneficial.

Overall, the results concerning log-ratio transformations inspire confidence and they itself do not seem to add additional difficulties.

10 PUBLICATION BIBLIOGRAPHY

Aitchison, J. (1982): The Statistical Analysis of Compositional Data. In *J. Royal Statistical Society* 44, pp. 139–177.

Aitchison, J. (2003): A Concise Guide to Compositional Data Analysis. Girona, Italy.

Althnian, Alhanoof; AlSaeed, Duaa; Al-Baity, Heyam; Samha, Amani; Dris, Alanoud Bin; Alzakari, Najla et al. (2021): Impact of Dataset Size on Classification Performance: An Empirical Evaluation in the Medical Domain. In *Applied Sciences* 11 (2), p. 796. DOI: 10.3390/app11020796.

Bailly, Alexandre; Blanc, Corentin; Francis, Élie; Guillotin, Thierry; Jamal, Fadi; Wakim, Béchara; Roy, Pascal (2022): Effects of dataset size and interactions on the prediction performance of logistic regression and deep learning models. In *Computer methods and programs in biomedicine* 213, p. 106504. DOI: 10.1016/j.cmpb.2021.106504.

Beleites, Claudia; Neugebauer, Ute; Bocklitz, Thomas; Krafft, Christoph; Popp, Jürgen (2013): Sample size planning for classification models. In *Analytica chimica acta* 760, pp. 25–33. DOI: 10.1016/j.aca.2012.11.007.

Coenders, Germa; Greenacre, Michael (2021): Three approaches to supervised learning for compositional data with pairwise logratios. Available online at http://arxiv.org/pdf/2111.08953v1.

Coenders, Germa; Pawlowsky-Glahn, V. (2020): On interpretations of tests and effect sizes in regression models with a compositional predictor. In *SORT 44*, pp. 201–220.

Gacesa, R.; Kurilshikov, A.; Vich Vila, A.; Sinha, T.; Klaassen, M. A. Y.; Bolte, L. A. et al. (2022): Environmental factors shaping the gut microbiome in a Dutch population. In *Nature* 604 (7907), pp. 732–739. DOI: 10.1038/s41586-022-04567-7.

Gloor, Gregory B.; Macklaim, Jean M.; Pawlowsky-Glahn, Vera; Egozcue, Juan J. (2017): Microbiome Datasets Are Compositional: And This Is Not Optional. In *Frontiers in microbiology* 8, p. 2224. DOI: 10.3389/fmicb.2017.02224.

Gordon-Rodriguez, Elliott; Quinn, Thomas P.; Cunningham, John P. (2021): Learning Sparse Log-Ratios for High-Throughput Sequencing Data.

Greenacre, Michael; Grunsky, Eric; Bacon-Shone, John (2021a): A comparison of isometric and amalgamation logratio balances in compositional data analysis. In *Computers & Geosciences* 148, p. 104621. DOI: 10.1016/j.cageo.2020.104621.

Greenacre, Michael; Grunsky, Eric; Bacon-Shone, John; Erb, Ionas; Quinn, Thomas (2022): Aitchison's Compositional Data Analysis 40 Years On: A Reappraisal. Available online at http://arxiv.org/pdf/2201.05197v1.

Greenacre, Michael; Martínez-Álvaro, Marina; Blasco, Agustín (2021b): Compositional data analysis of microbiome and any-omics datasets: a revalidation of the additive logratio transformation.

Hua, Jianping; Xiong, Zixiang; Lowey, James; Suh, Edward; Dougherty, Edward R. (2005): Optimal number of features as a function of sample size for various classification rules. In *Bioinformatics (Oxford, England)* 21 (8), pp. 1509–1515. DOI: 10.1093/bioinformatics/bti171.

Jasner, Yoel; Belogolovski, Anna; Ben-Itzhak, Meirav; Koren, Omry; Louzoun, Yoram (2021): Microbiome Preprocessing Machine Learning Pipeline. In *Frontiers in immunology* 12, p. 677870. DOI: 10.3389/fimmu.2021.677870.

Jiang, Ruochen; Li, Wei Vivian; Li, Jingyi Jessica (2021): mbImpute: an accurate and robust imputation method for microbiome data. In *Genome biology* 22 (1), p. 192. DOI: 10.1186/s13059-021-02400-4.

Kubinski, Ryszard; Djamen-Kepaou, Jean-Yves; Zhanabaev, Timur; Hernandez-Garcia, Alex; Bauer, Stefan; Hildebrand, Falk et al. (2022): Benchmark of Data Processing Methods and Machine Learning

Models for Gut Microbiome-Based Diagnosis of Inflammatory Bowel Disease. In *Frontiers in genetics* 13, p. 784397. DOI: 10.3389/fgene.2022.784397.

Lüll, Kreete; Arffman, Riikka K.; Sola-Leyva, Alberto; Molina, Nerea M.; Aasmets, Oliver; Herzig, Karl-Heinz et al. (2021): The Gut Microbiome in Polycystic Ovary Syndrome and Its Association with Metabolic Traits. In *The Journal of clinical endocrinology and metabolism* 106 (3), pp. 858–871. DOI: 10.1210/clinem/dgaa848.

Marcos-Zambrano, Laura Judith; Karaduzovic-Hadziabdic, Kanita; Loncar Turukalo, Tatjana; Przymus, Piotr; Trajkovik, Vladimir; Aasmets, Oliver et al. (2021): Applications of Machine Learning in Human Microbiome Studies: A Review on Feature Selection, Biomarker Identification, Disease Prediction and Treatment. In *Frontiers in microbiology* 12, p. 634511. DOI: 10.3389/fmicb.2021.634511.

Palarea-Albaladejo, Javier; Martín-Fernández, Josep Antoni (2015): zCompositions — R package for multivariate imputation of left-censored data under a compositional approach. In *Chemometrics and Intelligent Laboratory Systems* 143, pp. 85–96. DOI: 10.1016/j.chemolab.2015.02.019.

Pawlowsky-Glahn, Vera; Egozcue, Juan José (2016): Spatial analysis of compositional data: A historical review. In *Journal of Geochemical Exploration* 164, pp. 28–32. DOI: 10.1016/j.gexplo.2015.12.010.

Quinn, Thomas P.; Erb, Ionas (2020): Interpretable Log Contrasts for the Classification of Health Biomarkers: a New Approach to Balance Selection. In *mSystems* 5 (2). DOI: 10.1128/mSystems.00230-19.

Quinn, Thomas P.; Erb, Ionas; Richardson, Mark F.; Crowley, Tamsyn M. (2018): Understanding sequencing data as compositions: an outlook and review. In *Bioinformatics (Oxford, England)* 34 (16), pp. 2870–2878. DOI: 10.1093/bioinformatics/bty175.

Quinn, Thomas P.; Jacobs, Stephan; Senadeera, Manisha; Le, Vuong; Coghlan, Simon (2022): The three ghosts of medical AI: Can the black-box present deliver? In *Artificial intelligence in medicine* 124, p. 102158. DOI: 10.1016/j.artmed.2021.102158.

Rivera-Pinto, J.; Egozcue, J. J.; Pawlowsky-Glahn, V.; Paredes, R.; Noguera-Julian, M.; Calle, M. L. (2018): Balances: a New Perspective for Microbiome Analysis. In *mSystems* 3 (4). DOI: 10.1128/mSystems.00053-18.

Talwar, Divyanshu; Mongia, Aanchal; Sengupta, Debarka; Majumdar, Angshul (2018): AutoImpute: Autoencoder based imputation of single-cell RNA-seq data. In *Scientific reports* 8 (1), p. 16329. DOI: 10.1038/s41598-018-34688-x.

Topçuoğlu, Begüm D.; Lesniak, Nicholas A.; Ruffin, Mack T.; Wiens, Jenna; Schloss, Patrick D. (2020): A Framework for Effective Application of Machine Learning to Microbiome-Based Classification Problems. In *mBio* 11 (3). DOI: 10.1128/mBio.00434-20.

Tsamardinos, Ioannis; Rakhshani, Amin; Lagani, Vincenzo (2015): Performance-Estimation Properties of Cross-Validation-Based Protocols with Simultaneous Hyper-Parameter Optimization. In *Int. J. Artif. Intell. Tools* 24 (05), p. 1540023. DOI: 10.1142/S0218213015400230.

van der Ploeg, Tjeerd; Austin Peter C; Steyerberg Ewout W (2014): Modern modelling techniques are data hungry: a simulation study for predicting dichotomous endpoints. In *BMC Medical Research Methodology* 14, pp. 137–150.

Whalen, Sean; Schreiber, Jacob; Noble, William S.; Pollard, Katherine S. (2022): Navigating the pitfalls of applying machine learning in genomics. In *Nature reviews. Genetics* 23 (3), pp. 169–181. DOI: 10.1038/s41576-021-00434-9.

Wirbel, Jakob; Pyl, Paul Theodor; Kartal, Ece; Zych, Konrad; Kashani, Alireza; Milanese, Alessio et al. (2019): Meta-analysis of fecal metagenomes reveals global microbial signatures that are specific for colorectal cancer. In *Nature medicine* 25 (4), pp. 679–689. DOI: 10.1038/s41591-019-0406-6.

Zhang, Mo; Shi, Wenjiao (2019): Systematic comparison of five machine-learning methods in classification and interpolation of soil particle size fractions using different transformed data.

11 SUPPLEMENTARY

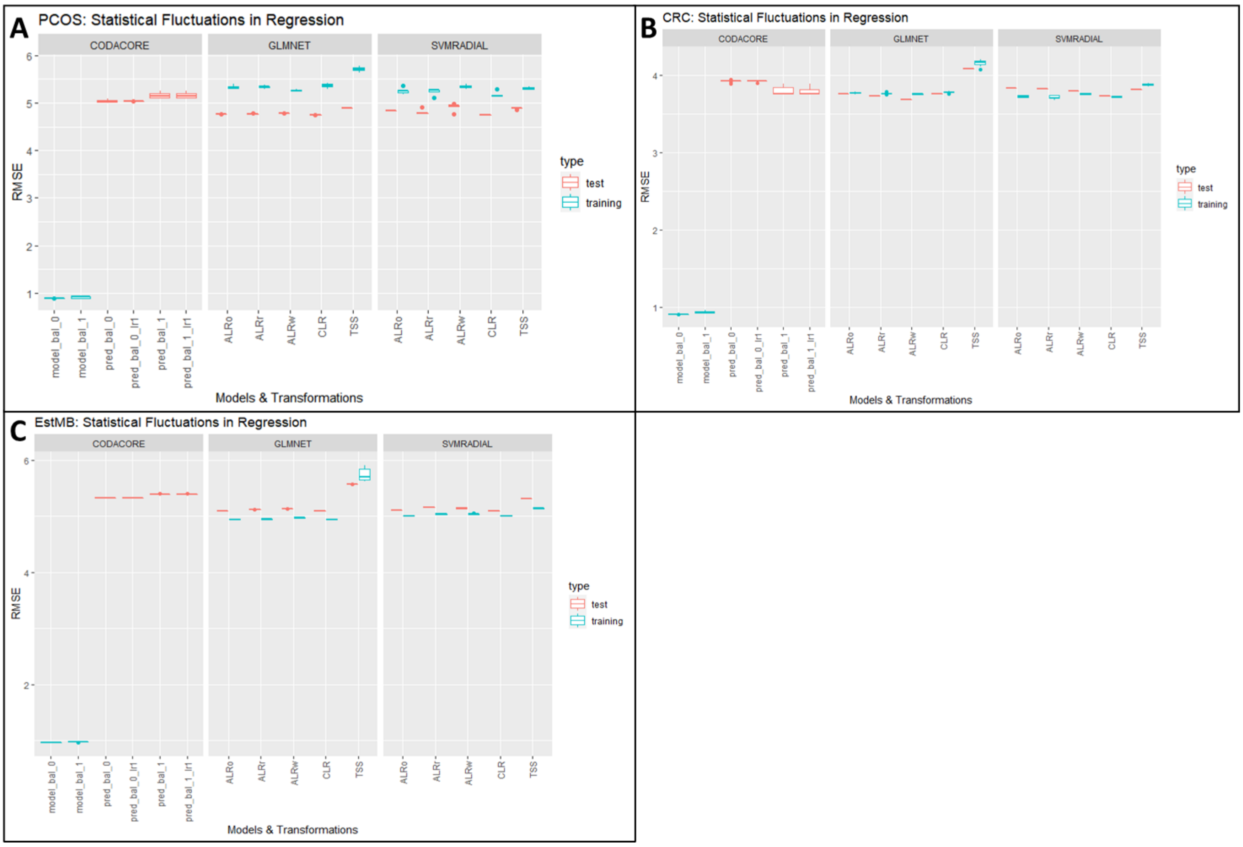

Figure 10: Statistical Fluctuations in Regression

(A) contains data from PCOS data set predicting BMI phenotype. (B) contains data from CRC data set predicting BMI phenotype. (C) and (D) both contain data from EstMB data set predicting BMI.

The y-axis contains AUROC scores. The x-axis contains models and transformations, with the most left always being *CoDaCoRe*, the middle portion displaying performances of glmnet and the right svmRadial. In red test performances are plotted and in blue training performances. model_bal_0 and model_bal_1 correspond to *CoDaCoRe* models with lambda 0 and lambda 1, respectively. Consequently, pred_bal_0 and pred_bal_0_lr1 correspond to predictions with all found log-ratios and only the most descriptive log-ratio for model_bal_0. The same nomenclature applies to pre_bal_1 and pred_bal_1_lr1. ALRo, ALRw and ALRr correspond to the types of ALR transformation applied in this pipeline (optimal, worst and random).

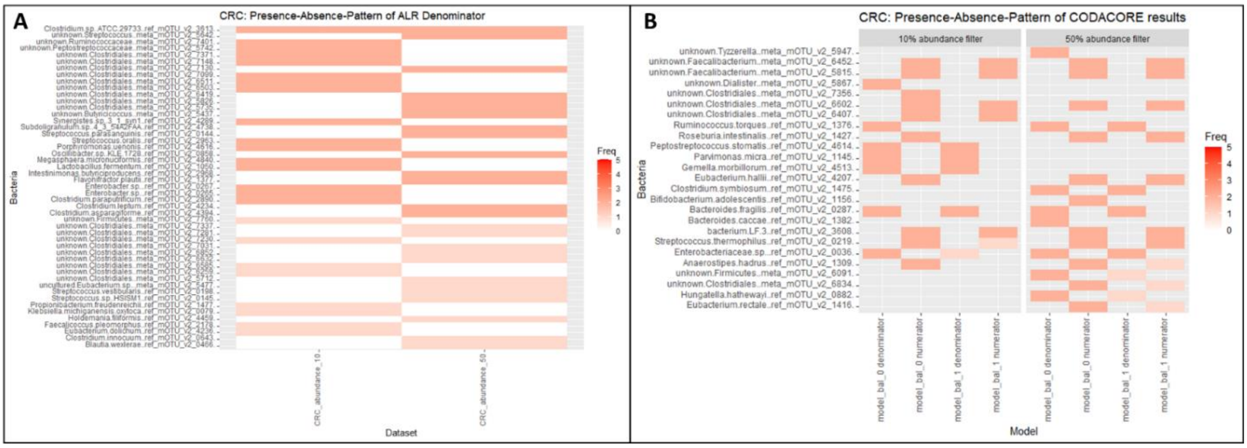

Figure 11: Presence-Absence-Patterns for CRC

(A) shows Presence-Absence-Pattern to show similarities and differences in calculated top 20 ALR references for CRC data set for differently sized data sets. (B) shows found numerators and denominators for both model types by CoDaCoRe also for differently sized CRC data sets.

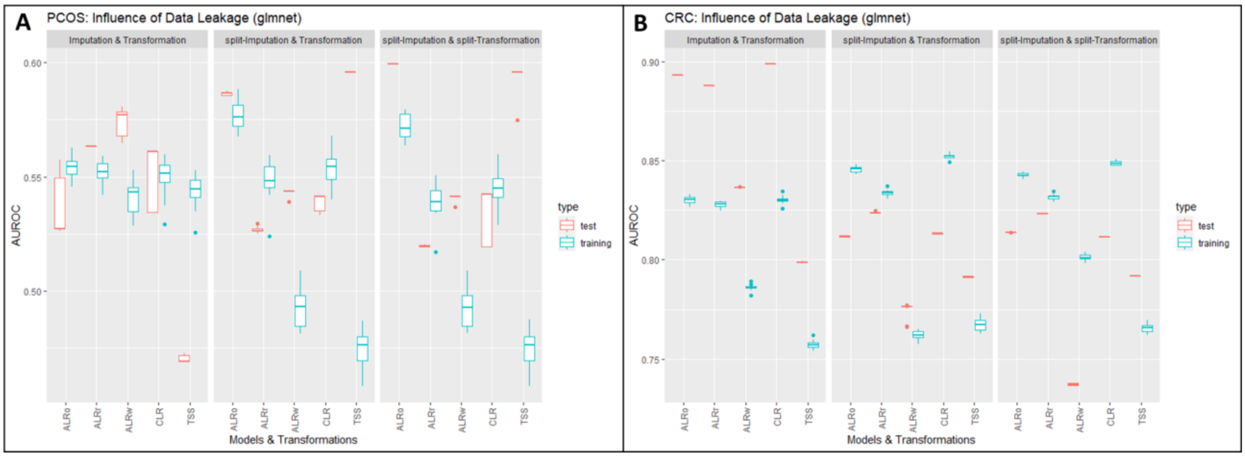

Figure 12: Influence of Imputation on Data Leakage

(A) contains data from PCOS data set predicting PCOS phenotype. (B) contains data from CRC data set predicting CRC phenotype. Left side shows Machine Learning performances for leaky-pipelines, in the middle data set was only split for Imputation and on the right side the complete non-leaky pipeline was applied.

The y-axis contains AUROC scores. The x-axis contains models and transformations, with the most left always being *CoDaCoRe*, the middle portion displaying performances of glmnet and the right svmRadial. In red test performances are plotted and in blue training performances. model_bal_0 and model_bal_1 correspond to *CoDaCoRe* models with lambda 0 and lambda 1, respectively. Consequently, pred_bal_0 and pred_bal_0_lr1 correspond to predictions with all found log-ratios and only the most descriptive log-ratio for model_bal_0. The same nomenclature applies to pre_bal_1 and pred_bal_1_lr1. ALRo, ALRw and ALRr correspond to the types of ALR transformation applied in this pipeline (optimal, worst and random).

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig und ohne fremde Hilfe angefertigt und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Die eingereichte schriftliche Fassung der Arbeit entspricht der auf dem elektronischen Speichermedium.

Weiterhin versichere ich, dass diese Arbeit noch nicht als Abschlussarbeit an anderer Stelle vorgelegen hat.

Declaration

I hereby declare that I have prepared this thesis independently and without outside assistance. I have not used any sources or aids other than those indicated. The submitted written version of the thesis corresponds to the one on the electronic storage medium.

Furthermore, I certify that this work has not been submitted as a thesis elsewhere.

Datum, Date

Unterschrift, Signature