MASTER'S THESIS

Normalization of metagenomic data
A comprehensive evaluation of existing methods

MIKAEL WALLROTH

Department of Mathematical Sciences
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2016

Thesis for the Degree of Master of Science

Normalization of metagenomic data
A comprehensive evaluation of existing methods

Mikael Wallroth

Department of Mathematical Sciences
Chalmers University of Technology and University of Gothenburg
SE – 412 96 Gothenburg, Sweden
Gothenburg, December 2016

Normalization of metagenomic data
A comprehensive evaluation of existing methods
MIKAEL WALLROTH

Supervisor: Mariana Buongermino Pereira, Department of Mathematical Sciences
Supervisor: Viktor Jonsson, Department of Mathematical Sciences
Examiner: Erik Kristiansson, Department of Mathematical Sciences

Master's Thesis 2016
Department of Mathematical Sciences
Division of Applied Mathematics and Statistics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Normalization of metagenomic data
A comprehensive evaluation of existing methods
MIKAEL WALLROTH
Department of Mathematical Sciences
Chalmers University of Technology

Abstract

Metagenomics is a growing area of bioinformatics, defined as the study of DNA sequenced from microbial communities rather than individual microorganisms. A particularly important part of metagenomics is to identify differences between communities, or conditions. This is typically done by statistical identification of differentially abundant features, such as genes or classes of genes, based on the number of times they have been observed in the samples. However, the data is high-dimensional and, due to undersampling, often sparse. Furthermore, it is well-known that metagenomic data is subjected to large technical variability, which unless taken into account, often makes correct inference hard.

When dealing with high-dimensional data, it is common practice to apply normalization methods to reduce the variance in the data. Several normalization methods of varying complexity have been suggested. The performance and characteristics of these methods have also been evaluated when applied to data from other fields, such as transcriptomics. However, to this day no comprehensive evaluation of methods have been done for metagenomic data. This, together with the fact that metagenomic data is characterized by large variability, makes a study of normalization methods of great interest.

In this thesis we consider nine commonly used normalization methods and analyze how they perform on metagenomic data in different situations. The methods are evaluated by studying how they influence the identification of differentially abundant features. Comparisons between methods are made by measuring how they affected the number of detected true and false positives in partly artificial data sets created by resampling of metagenomic data. The results presented in this thesis show that different methods may be preferred depending on condition sample size, the size of different abundance and how the highly abundant features are distributed between the conditions.

We conclude that several methods proven well-performing in related fields also perform well for metagenomic data. However, it is important to note that knowledge about the data and its effects is necessary, as different methods perform better in certain cases. Under extreme cases, normalization of metagenomic data can even result in a loss of statistical power.

Keywords: normalization, metagenomics, differentially abundant features, high dimensional data.

Acknowledgements

First, I would like to thank my supervisors, Mariana Buongermino Pereira and Viktor Jonsson, for their excellent guidance and the worthwhile discussions throughout this thesis. I would also like to thank my examiner Erik Kristiansson, for his large interest in the project and for his invaluable input. Finally, I would like to address my thanks to my family for being supportive and for encouraging me to pursue this degree.

Mikael Wallroth, Gothenburg, November 2016

Contents

1 Introduction **1**
- 1.1 Aim . . . 3
- 1.2 Delimitation . . . 3

2 Theory **5**
- 2.1 Differentially abundant features (DAFs) . . . 5
  - 2.1.1 Overdispersed Poisson generalized linear model . . . 6
  - 2.1.2 Analysis of variance and F-test . . . 7
  - 2.1.3 Multiple comparisons problem . . . 9
- 2.2 Normalization methods . . . 9
  - 2.2.1 Rate methods . . . 10
  - 2.2.2 Count methods . . . 13
  - 2.2.3 A brief take on similarities . . . 14

3 Methods **15**
- 3.1 Data . . . 15
  - 3.1.1 Artificial DAFs . . . 16
- 3.2 Experimental setup . . . 17
  - 3.2.1 Experimental parameters . . . 17
  - 3.2.2 Performance measures . . . 18

4 Results **21**
- 4.1 Performance evaluation on raw data . . . 21
- 4.2 Performance evaluation on data with artificial effects . . . 25
  - 4.2.1 Distribution of effects between conditions . . . 25
  - 4.2.2 Effect size . . . 27
  - 4.2.3 Fraction of affected features . . . 28
  - 4.2.4 Condition size . . . 29
- 4.3 Simulation of biological situations . . . 30

5 Discussion **35**
- 5.1 Simulated situations . . . 36
- 5.2 Similarities between methods and their assumptions . . . 39
- 5.3 Performance measures . . . 40
- 5.4 Conclusion . . . 40

Bibliography **43**

1
Introduction

Metagenomics is the study of the DNA found in environmental samples, representing communities of microorganisms [1]. The field of metagenomics has grown immensely in the last decade, much due to the availability of costly efficient sequencing techniques [2].

In traditional sequencing, DNA is extracted from a single organism, which typically implies that it is necessary to first isolate the target organism and subsequently clone it to obtain a sample large enough to extract the DNA from. One of the limitations of sequencing microbes arises from the fact that a very small fraction of the existing species can be isolated and clone *in vitro*. Metagenomics is the sequencing of microbial communities, where instead of sequencing from a single genome, the sequencing targets a collection of different genomes represented in a community. An advantage with metagenomics is that each organism already exists in several copies. Hence isolating and cloning may be omitted from the procedure of sequencing, and this in turn provides the ability of studying many species in a new way [3, 4].

A technique for sequencing metagenomic data is what is usually referred to as *shotgun metagenomic sequencing*. By randomly sequencing fragments of any of the genomes found in a sample, a large number of *reads* are generated. The reads, usually containing a few hundred nucleotides, may then be further processed, either by trying to align and reassemble the DNA, or by extracting information from each read. Another possibility is *targeted metagenomics*, where the sequencing targets some biological function, and hence only a subset of the genome is considered for sequencing. While the problem in this thesis is possibly transferable to target metagenomics, this thesis considers reads obtained through shotgun sequencing.

One common approach of studying the reads obtained from the shotgun sequencing is through binning. This procedure quantifies the set of reads by assigning each read to a bin, where each bin represents a *feature*. These features typically correspond to either operational taxonomic units (OTU), e.g. species, or functional units, e.g. genes [2, 5, 3].

By studying the relative abundance of such features across metagenomic samples, it is possible to distinguish key differences in the microbial communities, for instance between the microbial composition in the gut of healthy and ill patients [3]. The identification of *differentially abundant features* (DAFs) is however a difficult and complex problem. The data is usually very high dimensional with relatively few samples. Furthermore, biological data is often subject to high variation, and metagenomic data specifically is known to contain a large amount of both biological and technical variation [6].

Biological variability is caused by a natural biological diversity in terms of species

and gene content due to several factors, such as temperature, pH value and nutritive content. When studying differences between two or more conditions, we are interested in parts of this biological variability while the remaining variability is something we simply have to accept to exist in the data. The studied microbial communities are also highly dynamical systems, and every sample contains by itself high variations over time, which are not capturable by a single sequencing, thus representing another kind of biological variation [2]. In short, some variation in the data is expected to be there, both between conditions and between samples within the same condition, but it may be troublesome to distinguish it from technical variations.

Technical variations include for instance bias caused by equipment and/or experiment, *sequencing depth*, sequencing quality and databases. The sequencing depth defines the number of random fragments generated and sequenced from a sample, which is the cause of a large part of the technical variations. The reason is that biological variation may arise from important, yet rare organisms that exist in only one of the conditions, which may not be present in all samples corresponding to the environment where it is present. Furthermore, the rarity of some organisms implies that they simply may not be observed due to a limited sequencing depth [7]. In other words, the sequencing depth may sometimes be insufficient to capture the whole environment in all samples, which manifests itself as large variations in the data. One way of reducing this variation is to increase the sequencing depth, as this would increase the probability of the observed DNA content. However, the sequencing depth is a trade-off, as sequencing is still an expensive procedure, and at the same time the sequencing depth is fixed once the data is sequenced, and cannot be increased if found insufficient, unless the sequencing is redone. In short, the technical variability can be substantial and is subject to minimization, as it impairs inference made about the biological variability [8].

To control variation one may use normalization methods, for which the purpose is to bring the samples to a common scale and hence make samples comparable. Normalization is proven to play a significant role in the differential analysis when considering microarrays as well as RNA-seq data, both for which several methods have been proposed and compared [9, 10]. To this day there still exists a lot of uncertainty regarding normalization of metagenomic data. While many methods used for RNA-seq data are also directly applicable to metagenomic data, there is no guarantee that all methods perform equally well for both type of data. For instance, while the earlier mentioned types of variation appear in most biological data, the actual properties of these variations may differ between data types. Hence a method that performs sufficiently well for one data type may in another case prove insufficient.

What is known however is that normalization is still essential for metagenomic data, and further the choice of normalization method has a notable effect on the down stream analysis [11]. Also, while several normalization methods from other related types of data are directly applicable to metagenomic data as well, such as trimmed mean of M-values [12], there exist methods that have been suggested specifically for metagenomic data, for instance cumulative sum scaling [7].

In this thesis we compare nine normalization methods that been used for metage-

nomic or other closely related data, and evaluate the performance of the the methods when applied to metagenomic data.

1.1 Aim

The aim of this thesis is to

- theoretically describe and analyze the most common and important normalization methods used for metagenomic data or closely related data,
- implement the normalization methods and evaluate their performance in terms of removing systematic errors in metagenomic data,
- investigate and explain the behaviour of the normalization methods in different experimental situations.

1.2 Delimitation

The problem of identifying differentially abundant features in metagenomic data is vast in the sense that there exist numerous factors that contribute to the down stream analysis. As this thesis treats normalization methods applicable to the problem, it should be noted that there are a few noteworthy limitations. For instance, the results in this thesis are based on data from mainly one data set, which may not be fully representative to metagenomic data in general.

Furthermore, there exist several methods for identification of differentially abundant features, or closely related problems, but this thesis will only treat one such method. While we may expect that the results should be comparable, statistical methods might differ in what normalization is most suitable, as there are always assumptions included that could influence results in one way or another.

Lastly, this thesis considers nine normalization methods that are considered to be popular or otherwise interesting to include. It should be noted that there exist other methods not included in this thesis, that have been suggested for either metagenomic data or related data.

2

Theory

In this thesis the problem of identifying differentially abundant features (DAFs) is studied. More specifically, the thesis treats the implications of normalization of data prior to the analysis. In this chapter we introduce basic notion related to differential abundance, we describe how feature abundance is modeled and how *differential* abundance can be identified from a statistical point of view. Furthermore, this chapter describes the different normalization methods in depth.

2.1 Differentially abundant features (DAFs)

Consider multiple metagenomic samples being sequenced according to a shotgun metagenome sequencing procedure, with each sample providing a set of reads, i.e. random short fragments of the full genome. As each of these reads may be incomplete in the sense that, for instance, it may not contain a full gene, the approach is to instead use techniques for clustering reads based on similarities. This procedure is referred to as *binning*. By doing so, each read is assigned to the bin it is most similar to and each bin corresponds to a *feature*, which in turn are operational taxonomic units (OTUs) (eg. species) or functional units (eg. genes), and instead of observing raw sequence fragments we have counts of observations for each feature.

The process of binning results in the *feature abundance matrix*, denoted Y, where each row corresponds to different features and each column is a sample, i.e. an individual or an environmental sample. Hence the element Y_{ij} is the number of observations of feature i in sample j [4]. Considering that each sample belongs to one of two *conditions*, for instance healthy and diseased, a feature is said to be *differentially abundant* if its abundance differs notably between two conditions.

There exist several methods and tools implemented that may be used to determine whether the feature abundance is significantly different between conditions. Popular tools are edgeR [13], DESeq2 [14] and MetagenomeSeq [7], to name a few. However, for simplicity we will restrict ourselves to an overdispersed Poisson generalized linear model (oGLM), which has been proven to perform comparable to the other methods [15]. The model is described in detail in section 2.1.1, but in short the oGLM models the feature abundance Y_{ij} with a Poisson distribution and the expectation of feature abundance by a linear combination of predictor variables dependent on the condition of each sample, and links the predictor with the expected feature abundance through a logarithmic function. Note that this thesis treats the normalization methods rather than methods for identifying DAFs.

2.1.1 Overdispersed Poisson generalized linear model

A generalized linear model is a model that links A to B using C, and it is defined by three components [16, 17], namely:

- *Random component* – the response variables $\mathbf{Y}$, where $\mathbf{Y}$ is $n \times 1$ and n the number of observations, are called random components. While not identically distributed, they are all distributed according to some exponential family, e.g. normal, Poisson or binomial.
- *Systematic component* – the linear predictor $\boldsymbol{\eta} = \boldsymbol{\beta}\mathbf{X}$ is called the systematic component, and is the model obtained by the matrix of predictor variables $\mathbf{X}$, where $\mathbf{X}$ is $n \times p$ and $\boldsymbol{\beta}$ is $p \times 1$, where p is the number of predictor variables in the model.
- *Link function* – the link function f connects the random component with the systematic component by describing the relation $f(\boldsymbol{\mu}) = \boldsymbol{\eta}$, where $\mu = \mathbb{E}[\mathbf{Y}|\mathbf{X}]$. Here f depends on the assumed distribution family.

In our case we will consider a generalized linear model for each feature, hence, for each feature i and sample j we model the the mean of each feature abundance Y_{ij} as

$$\mathbb{E}[Y_{ij}|\mathbf{x}_j] = f^{-1}(\boldsymbol{\beta}_i\mathbf{x}_j)$$

where the link function f depends on the assumed exponential family. In the case of Poisson regression, the link is defined by the logarithm, i.e. we have

$$\log(\mathbb{E}[Y_{ij}|\mathbf{x}_j]) = \boldsymbol{\beta}_i\mathbf{x}_j$$

The predictor variables used depend on the assumptions of the model. The first assumptions in our model is that counts may be differentially abundant between conditions, which is translated into the systematic component defined as $\eta_i = \alpha_i + \beta_i x_j$, where x_j is the indicator function dependent on the condition, i.e.

$$x_j = \begin{cases} 1, & \text{if sample } j \text{ belongs to condition 1} \\ 0, & \text{if sample } j \text{ belongs to condition 2.} \end{cases}$$

Here α_i is called the intercept and is functioning as a baseline, while β_i is the slope and describes the relative abundance between conditions. The Poisson model is however not sufficient for modelling our problem, as it assumes that the counts are on a common scale. Instead, we want to extend this count model to a rate model where, instead of expected counts, we model the counts as a fraction of some measure that describes the whole sample, such as the total number of counts. We can get a rate model by adding a sample-dependent predictor $\log(N_j)$ that represents a scaling factor that normalizes the data.

We will refer to N_j as the *normalization factor* and estimating this constant may be done by applying normalization methods. It should be noted that some of the normalization methods in this thesis may not describe the normalization by a scaling factor, but instead relies on computing a new feature abundance matrix $\bar{Y}$. The two types of normalization approaches are more thoroughly described in section 2.2. For

now, we conclude that with the inclusion of the predictor $\log(N_j)$, we have arrived to the full model, i.e.

$$\log(\mathbb{E}[Y_{ij}|x_j]) = \alpha_i + \beta_{ij}x_j + \log(N_j).$$

The model may equivalently be written as

$$\log\left(\frac{\mathbb{E}[Y_{ij}|X]}{N_j}\right) = \alpha_i + \beta_{ij}x_j \Leftrightarrow \mathbb{E}[Y_{ij}|X] = N_j \exp(\alpha_i + \beta_{ij}x_j), \tag{2.1}$$

which highlights how adding the normalization factor transforms the count model to a rate model. Using our data, i.e. the known counts Y_{ij}, the conditions x_j and estimated normalization factors N_j, we fit the model by estimating α_i and β_i. As earlier mentioned our assumption of this model is that the counts are distributed according to a Poisson distribution with overdispersion. The reason is that the Poisson model is rather strict as it carries the property that $\text{var}(Y) = \mathbb{E}[Y]$. However, metagenomic data often has larger variation than modelled by the Poisson distribution, and thus by allowing for overdispersion, i.e. $\text{var}(Y) > \mathbb{E}[Y]$, we may expect a more reliable fit for our model. In practice, allowing for overdispersion implies that we model the variance through some function other than identity. The overdispersed Poisson model used is also called the quasi-Poisson, where we have that

$$\mathbb{E}[Y_{ij}] = \mu_i, \quad \text{var}[Y_{ij}] = \phi_i\mu_i, \quad \phi \geq 1.$$

Here we note that $\phi = 1$ would indeed correspond to the regular Poisson model. Thus, for $\phi > 1$, we have a larger variance than the one found in a regular Poisson model, which is adequate to deal with metagenomic data that carry large variation.

The main difference caused by the overdispersion is how the model is fitted. In a regular Poisson regression model, we fit the model, i.e. estimate α_i and β_i by maximum likelihood estimates (MLE). However, when mean and variance are not equal the MLE has generally no analytical expression [18]. Instead it is possible to use a technique called Iteratively reweighted least squares (IRLS) to find the MLE. IRLS computes a weighted least square expression and updates the weights, and repeats until the change in the weights between iterations is sufficiently small [16].

The oGLM described above is the foundation for evaluating the different normalization methods described in section 2.2. As mentioned at the beginning of this chapter, the model is a practical choice since it is easy to understand from a mathematical point of view. This becomes obvious through equation 2.1 as it clearly shows the role of the normalization factors N_j in modelling the abundance. While the model describes the abundance, we will now describe what is considered *differential* abundance in our model.

2.1.2 Analysis of variance and F-test

When a feature i is differentially abundant between two conditions the predictor coefficient β_i in the oGLM is different from zero. This is because of β_i defining the

relative abundance, and hence if the feature truly is differentially abundant, this should be expressed by β_i being non-zero. Thus, testing for differential abundance translates into testing if β_i is significantly different from zero. In other words, let the null hypothesis be defined as $H_0 : \bar{\beta}_i = 0$, where $\bar{\beta}_i$ is the true value on the slope constant, and subsequently the alternative hypothesis $H_1 : \bar{\beta}_i \neq 0$. We proceed by considering a pair M_0 and M_1 of nested models:

$$M_0 : \log\left(\frac{\mathbb{E}[Y_{ij}]}{N_j}\right) = \alpha_i,$$
$$M_1 : \log\left(\frac{\mathbb{E}[Y_{ij}]}{N_j}\right) = \alpha_i + \beta_i x_j$$

where M_0 is known as the reduced model and M_1 as the full model. It should be clear that the two models correspond to the null hypothesis H_0 and alternative hypothesis H_1, as M_0 is equivalent to $\beta_i = 0$, and M_1 analogously $\beta_i \neq 0$. The test is essentially which model fits best, or whether M_1 is considerably more beneficial than M_0.

When comparing models with overdispersion where the variance is larger than the expectation and therefore unknown, it is appropriate to use an F-test, instead of a Likelihood ratio test which assumes the variance to be known [19, 20]. In the F-test we essentially study the statistic:

$$F = \frac{\text{variance between groups}}{\text{variance within groups}}$$

or more formally,

$$F_i = \frac{\frac{(SSE_{i,0} - SSE_{i,1})}{df_0 - df_1}}{\frac{SSE_{i,1}}{df_1}}.$$

Here $SSE_{i,k}$ denotes the residual sum of squares for feature i and model k, i.e.

$$SSE_{i,k} = \sum_{j=1}^{n} (f_{i,k}(x_j) - Y_{i,j})^2,$$

where $f_{i,k}(x)$ are the regression models defined by $f_{i,0}(x) = \alpha_{i,0}$ and $f_{i,1}(x) = \alpha_{i,1} + \beta_{i,1}x$, and x_j is the indicator function dependent on the condition, i.e.

$$x_j = \begin{cases} 1, & \text{if sample } j \text{ belongs to condition 1} \\ 0, & \text{if sample } j \text{ belongs to condition 2.} \end{cases}$$

Furthermore, df_k are the degrees of freedom for the two models, here $df_0 = n - 1$ and $df_1 = n - 2$. Our F-statistic F_i is said to follow the $F_{(df_0 - df_1, df_1)}$ distribution under the null hypothesis H_0. The p-value is then simply equal to the probability of a $F_{(df_0 - df_1, df_1)}$-distributed random variable X to be greater or equal to F_i, i.e.

$$p_i = \mathbb{P}(X \geq F_i).$$

We lastly compare the p-values against a significance level α. As the p-values are to be interpreted as the probability of obtaining F under H_0, we consider p-values smaller than α to be significant in the sense that it is very unlikely that the null hypothesis H_0 is true.

2.1.3 Multiple comparisons problem

The significance level, α, defines the probability of falsely rejecting the null hypothesis. Due to high dimensionality in the metagenomic data, some of these unwanted null hypothesis rejections are expected to occur just by chance, which is commonly referred to as the multiple comparisons problem. One approach of controlling this problem is the Bonferroni method, where H_0 instead is rejected if $p < \alpha/m$, where m is the number of comparisons. This procedure is however generally too conservative for large m, in the sense that α/m becomes very small which results in very few rejections. Instead we use the Benjamini-Hochberg correction [21]. This procedure adjusts the p-values by controlling the false discovery rate (FDR). Let p_i, $i = 1, \ldots, m$ be the obtained p-values, and $p_{(i)}$ be the ordered p-values. Further, let k be the largest integer $k \in [1, m]$ such that

$$p_{(k)} \leq \frac{k}{m}\alpha.$$

Benjamini-Hochberg then states that we reject H_0 for all p-values smaller than $p_{(k)}$. This is equivalent to have adjusted p-values, denoted by q_i, given by:

$$q_{(i)} = \frac{m}{i}p_{(i)},$$

and subsequently reject H_0 if $q_i < \alpha$. The argument behind Benjamini-Hochberg correction is to control the false discovery rate (FDR). We define the FDR as

$$FDR = \mathbb{E}\left[\frac{D_F}{D}\right],$$

where D is the number of discoveries, i.e. the number of times we rejected H_0, and D_F is the number of false discoveries, i.e. the times H_0 was rejected when H_0 was in reality true. By using the Benjamini-Hochberg correction, the p-values are adjusted so that the estimated FDR is equal to the signifiance level α.

2.2 Normalization methods

As earlier described, the main purpose of this thesis is to evaluate different normalization methods when applied to metagenomic data. Before describing each method, it should be noted that there are two general groups of normalization methods, namely *rate methods* and *count methods*.

Rate methods computes the normalization factor N_j in the generalized linear model, as described in the previous section. This approach, describing the data as a rate relative to some sample specific constant, is the most commonly used in the studied methods.

Meanwhile, count methods normalize by generating a new data set using information given by the original data set, and instead perform the differential abundance analysis on this data set. This implies that we consider the data as counts rather than rates.

2.2.1 Rate methods

Total count (TC): Total counts is one of the methods that scales counts into rates, as such the counts for each feature is divided by a normalization factor N_j that in this case is library size, i.e. the sum of counts of all feature in that sample, that is

$$N_j = \sum_{i=1}^{m} Y_{ij}.$$

In terms of the oGLM, this suggests that the model now describes the counts as proportions of the sample. Because the library size, or total counts per sample, is directly related to the sequencing depth, this method relies on the assumption that the single largest noise in the data is the difference in sequencing depth, which may differ in orders of magnitude.

A drawback of this method is that by accounting for all features when normalizing, we introduce an error caused by the differentially abundant features. That is, since our assumption is that we actually expect to find some significant differences between conditions, we introduce a bias by not including this assumption in the normalization. The size of this error depends on the size and frequency of differentially abundant features. It may be argued that a more viable approach is to use quantile based normalization methods [22]. The reasoning here is that while the proportions may be misleading, the distribution of counts should be roughly equal across all samples.

Median (Med): Normalizing by the median is as straightforward as normalizing by total counts, here instead, we define the normalization factor as

$$N_j = \underset{i \in G^*}{\text{median}}\, Y_{ij}, \quad G^* = \{i : \sum_j Y_{ij} > 0\},$$

that is the sample-wise medians of all features with at least one non-zero count [9]. While the total count method is expected to be sensitive to outliers, the median is known to be resistant to outliers. In our data, outliers may for instance correspond to differentially abundant features or just features that in rare cases has very high counts. Either way, they are of little interest when normalizing, as the purpose is to equalize those features that should be close to equal across samples.

However, it should be mentioned that an excess of zero-count or low-count features could influence this method in a negative manner, as the median will become less informative about the non-zero counts and is likely less representative of the high-count features in these situations [22].

Upper quartile (UQ): The upper quartile method is another rate method, here we divide the counts for each feature in the same sample j by the upper quartile (75th percentile), which is our normalization factor N_j, i.e.

$$N_j = \underset{i \in G^*}{\text{upper quartile}}\, Y_{ij}, \quad G^* = \{i : \sum_j Y_{ij} > 0\}.$$

The motivation of using the upper quartile is similar to that of the median, i.e. that it is a method robust to outliers, but an argument of upper quartile being superior

to median is that the upper quartile should be less sensitive to zeros in the data, and subsequently a more stable statistic to normalize by [22]. This is due to the fact that even with many zero-count and low-count features, the upper quartile is expected to better reflect the distribution as it considers higher count features, while still not the most extreme counts. It is however important to note that this method assumes that the underlying distribution is approximately equal for all samples, and hence may still be inaccurate in situations where the difference in count distribution between the samples is too large.

Trimmed mean of M-values (TMM): TMM normalization was suggested as a robust alternative approach to normalizing by the total counts, and is included in edgeR package. The TMM method normalizes the data by initially choosing a reference sample r from the available samples. The idea is then to estimate one normalization factor N_j for each sample j by an adjusted library size relative to a reference sample r, i.e.

$$N_j = f_j^{(r)} \sum_{i=1}^{n} Y_{ij}$$

where the adjustment of the reference sample relative to itself is $f_r^{(r)} = 1$. $f_j^{(r)}$ is estimated as follows. First,given the counts Y_{ij} and Y_{ir} for feature i in sample j or reference sample r and library size N_j and N_r of sample j or reference sample r, we define the log-fold change M_{ij} and the absolute expression A_{ij} as

$$M_{ij}^r = \log_2 \left(\frac{Y_{ij}/N_j}{Y_{ir}/N_r} \right)$$
$$A_{ij}^r = \frac{1}{2} \log_2 \left(\frac{Y_{ij}}{N_j} \frac{Y_{ir}}{N_r} \right).$$

Here the indices are to be interpreted as sample $j \in J$ relative the reference sample r, for feature $i \in G$. We form the reduced set of features G^* by trimming the set G, i.e. removing the top and bottom extremes, based on the M-values and A-values, by default 30% and 5% respectively. As the M-values correspond to the log-fold change, trimming the extremes of M-values implies that we do not consider the features with the largest relative abundance when normalizing. Similarly, trimming by the A-values implies that we exclude the features with the most extreme absolute abundance.

Next, we define the library scaling constant f_j^r by

$$\log_2(f_j^{(r)}) = \frac{\sum_{i \in G^*} w_{ij}^r M_{ij}^r}{\sum_{i \in G^*} w_{ij}^r}$$
$$w_{ij}^r = \frac{N_j - Y_{ij}}{N_j Y_{ij}} + \frac{N_r - Y_{ir}}{N_r Y_{ir}}$$

for $Y_{ij}, Y_{ir} > 0$. The above equations state that the $\log_2(f_j^{(r)})$ is defined as the weighted average of the trimmed M-values. Here the weights used are the inverse of the variance, computed through the delta method. The weights are used to account

for that the log-fold changes from high count features have lower variance on a logarithmic scale.

The assumption in this method is that only a few features are differentially abundant [12, 9, 23]. This assumption is expressed in that f_j ideally should be equal to 1, or equivalently, $\log_2(f_j^{(r)}) = 0$. When f_j^r is equal to 1, the normalization constant N_j is equal to the library size, i.e. the same normalization constant obtained through the TC method.

This assumption is expressed by that a data set with few DAFs should have a f_j^r close to 1 for all samples j, implying that the library size, i.e. total counts, is an accurate normalization factor. If f_{ij}^r deviates from 1, then adjusting each library size by its corresponding constant should ensure that the assumption holds.

DESeq: This method is proposed by the creators of the DESeq package, and hence referred to by package name. The idea of this method is to form a reference sample by the geometric mean across samples, and define the normalization constants as the median of the ratios between sample and reference sample. That is, the normalization factor is in this case computed as

$$N_j = \text{median}_i \frac{Y_{ij}}{\left(\prod_{j'=1}^{m} Y_{ij'}\right)^{1/m}}.$$

As with TMM, the hypothesis of this method is that only a few features are differentially abundant, and thus non-DAFs should have similar read counts across samples. Hence, the ratio between the counts and geometric mean should ideally be equal to 1 for non-DAFs. The assumption is then that the median of the ratios should serve as a stable estimate of the normalization factor. [24, 9].

Cumulative sum scaling (CSS): The cumulative sum scaling is a rate method that computes the normalization factor as a sum over a subset of the features [7]. More specifically, the normalization factor is computed as

$$N_j = \sum_{i:Y_{ij} \leq q_j^{\hat{l}}} Y_{ij},$$

where q_j^l is the $\hat{l}$th quantile of the jth sample, with $\hat{l}$ determined through the following steps:

1. Define $\bar{q}_l = \text{median}_j q_j^l$.
2. Define $d_l = \text{median}_j |q_j^l - \bar{q}_l|$.
3. Then $\hat{l} = \min\{l : d_{l+1} - d_l \geq c d_l\}$,

i.e. the smallest l under which the inequality holds. Here c is a constant specifying the relative bound of when instability occurs, by default $c = 0.1$. The assumption of which CSS is based upon, is that the count distributions should be approximately equivalent up to a certain quantile. The authors of this method argue that by estimating this quantile and normalizing by the total counts of this subset, the samples should be brought to a common scale.

Reversed cumulative sum scaling (RCSS): The RCSS is an alternative approach to CSS, where instead of determining the normalization constants by the sum over all counts smaller than some quantile, the normalization constants are determined by the sum over all counts larger than some quantile, typically in the upper half of the counts. Unlike CSS, where the exact quantile is dependent on how large the variations are in each quantile and may vary for different data sets, we let the quantile in RCSS be fixed.

2.2.2 Count methods

Quantile (Q): The quantile normalization is a count method, thus, the normalization will replace counts by normalized counts instead of scaling to a rate as in rate methods. The idea of the quantile normalization is to fix the distribution of feature counts across samples to be identical, which is done by computing the median of quantiles across samples, and normalizing according to it. The median quantile is then

$$\bar{q}_l = \underset{j \in S}{\text{median}}\, q_{lj},$$

where q_{lj} is the lth quantile in the jth sample. Here we include an exception for ties occurring, be it due to an even number of samples or identically valued counts when computing the quantiles, both of which are handled at random. The reason of this is to ensure that the data are still considered counts after normalization. The counts Y_{ij} is then replaced according to the quantiles, i.e. we replace Y_{ij} by $\bar{Y}_{ij}$ such that $q_{lj} = \bar{q}_l$.

It should be noted that the quantile normalization was argued to be one of the best normalization methods for microarray data [10], and while some modifications were made to handle count type data, the idea is essentially the same.

Rarefying Rarefying is a method similar to the TC method, as it assumes that the sequencing depth is the largest source of technical variation, and like TC, rarefying ensures that all samples have the same sequencing depth. However, rarefying is a count method and not a rate method, as instead of scaling by the library size, rarefying implies that each sample is reduced to the smallest library size by removing counts at random. That is, all samples are considered as if they were all sequenced to the same depth from the start.

While the reasoning behind rarefying may at first sound reasonable, it should be noted that the method practically implies dismissing a lot of information given by the raw samples. It has been shown that rarefying undermines the performance of subsequent analysis [11], due to added noise. Further, the variance is flattened by increasing rather than decreasing it, which generally is not a preferable property in normalization methods. Yet, the approach exists and is used, and hence it is of interest to be included in a study of normalization methods.

2.2.3 A brief take on similarities

At a glance, these methods are both similar and different in several ways. Without any deeper arguments, the similarity between Med and UQ is obvious and already mentioned. Furthermore, TC, Med, UQ and RCSS consider all samples to be independent, while TMM, quantile, DESeq, CSS and rarefying use, in one or another way, information given by the collection of samples, for instance by forming a reference sample or simply by rarefying.

Similarities arise from the assumptions and ideas behind each method. For instance, an important assumption is that most features are not differentially abundant, which is the basis of TMM and DESeq. Another general idea is assumptions regarding that all samples in some way share the same underlying distribution. This assumption is used in Med, UQ, TMM, Quantile, DESeq, CSS and RCSS, but to different extents. While Med and UQ equalizes one quantile across all samples, CSS and RCSS assume that a subset of the counts are equally distributed, and Quantile says that each quantile should be equal across samples.

A more specific similarity to consider is that the reference sample used in CSS is the same as the one used in the quantile normalization method [7], with the exception that quantile normalization ensures that the reference samples are still counts. However, while quantile normalization uses the reference sample directly, CSS only uses the sample as a reference when estimating how large the spread is in each quantile.

Similarities such as these are a vital part of analysing the performance of methods, partly in terms of understanding why some methods performs well, but also which concepts to consider when developing and improving normalization methods in the future.

3
Methods

In this thesis we want to evaluate how different normalization methods perform under the problem of identifying differentially abundant features in metagenomic data. In this chapter the procedures for evaluating the different normalization methods are described. Furthermore, we briefly explain problems of measuring performance and explain the approach to circumvent some of these problems.

3.1 Data

Ideally, we want to measure performance using real metagenomic data where all differentially abundant features (DAFs) are already known, and compare the ones found by our model with the true DAFs, as this would best represent a real life situation. However, as we lack real metagenomic data with known DAFs, we have a trade-off situation where we may consider real data with unknown DAFs, or simulated data with known DAFs. Real data would imply that we cannot evaluate the correctness, and instead we could obtain a set of DAFs without being able to verify their correctness. On the other hand, simulating data from a probability distribution also has its limitations, as it implies that we force assumptions on the data that may or may not be there. Furthermore, with simulated data it may be hard to fully capture the large variation of the real data.

Instead, a compromising, viable approach is to use a non-parametric bootstrap. The idea is to use real data so that we use its variation and inforcing less assumption regarding the distribution of the data. A sufficiently large bootstrapping should remove the DAFs present in the samples, which implies that we can add artificial DAFs that are known to us, thus making it possible to measure performance of the normalization methods. Non-parametric bootstrap is a basic Monte-Carlo procedure where we iteratively create new data sets by randomly drawing data sets with replacement. In our case, we choose m samples among all the samples in one of our two conditions. Next, the m samples are then assigned to artificial conditions, with m_1 and m_2 samples in each artificial condition, so that $m_1 + m_2 = m$. As we iteratively scramble all samples from one of the initial conditions, pick m samples randomly with replacement and assign them to one of two artificial conditions, the null hypothesis is considered true for a sufficiently large number of iterations, i.e. we may consider that there will be no DAFs present in the data set. We may then subsequently add artificial DAFs to the bootstrapped data, as explained next.

3.1.1 Artificial DAFs

Since for a large enough resampling on the same condition, we can assume the bootstrap will remove in average DAFs that randomly occur, we then add artificial *effects*, i.e. features that are more abundant in one of the conditions, to the data sets that represent DAFs that we know and can control. The advantage of this is that since we manually add effects, we also know which features that are truly differentially abundant in our experiment.

If we consider the bootstrapped data set, we have n features and m samples. We assign each sample randomly to one of two artificial conditions such that both conditions are represented by equally many samples. To create artificial DAFs, we add effects to a fraction of the features, for instance 10% of the n features, and only to samples in one of the conditions, generally not the same condition for all effects. For each affected feature i, and each sample j in the affected condition, we replace the counts Y_{ij} by new counts taken from a binomial distribution, as follows:

$$\hat{Y}_{ij} \sim \text{Binomial}(Y_{ij}, p),$$

where p is the *effect size*, i.e. one over the abundance fold change.

We consider two ways of adding effects by downsampling. The first is what we call *balanced* which is done by adding equally many effects to each condition. That is half of the artificial DAFs are created by downsampling counts in the first condition, and the rest created by downsampling counts in the second condition. This represents a situation where it is reasonable to assume that a function lost in one environment is replaced by another, in other words, the low abundance of one feature leaves room for another feature to have high abundance. The second way is what we call an *unbalanced* situation, where all downsampling is made to one of the conditions. This situation could occur in target studies, where you aim to sequence a set of genes related to one function, such as the abundance of antibiotic resistant genes in pristine or polluted environments; in such cases, when only a set of genes is studied, they are expected to be found in one condition and not in the other, which leads to an unbalanced presence of overly abundant genes in only one condition.

Adding effects through downsampling changes the total number of counts in the samples. In a balanced situation we expect that this change is equal for both conditions. However, for the unbalanced case we introduce a condition dependent change to the total counts, i.e. an artificial difference in the total counts between the two conditions. To counteract this all the features in the unchanged condition are randomly downsampled in order to keep the column totals unchanged. That is, if we downsample x% features in the first condition according to the above rule, we also downsample all features in the second condition according to

$$\begin{aligned}\hat{Y}_{ij} &\sim \text{Binomial}(Y_{ij}, p_2),\\ p_2 &= 1 - (1 - p_1)xc_1,\end{aligned}$$

where c_1 is the fraction of effects in condition 1 out of the total number effects. If we also add effects to condition 2, the counts in condition 1 are downsampled analogously.

3.2 Experimental setup

The data was run for several different values on the *experimental parameters*, which are condition size, effect size, balancing and fraction of affected features, with a total of 48 combinations. For each combination of experimental parameters 100 bootstrap iterations were performed. In each such iteration, we

1. added effects,
2. computed the normalization constants for the current data, as explained in section 2.2,
3. performed the identification of DAFs using the overdispersed Poisson model described in section 2.1, with each of the normalization constants in step 2.

It should be noted that all normalization methods were run with default options where applicable. Methods such as TMM and CSS have options that are highly data dependent, and adjusting these options could of course alter the results for better or worse. Any such studies was however not considered in this thesis.

3.2.1 Experimental parameters

In this simulation we considered the four *experimental parameters*, namely condition size, effect size, balancing and fraction of affected features. Since a large part of the results presented in chapter 4 focuses on how the experimental parameters affect the different normalization methods, each parameter is here briefly defined.

Condition size The condition size determines how many samples we consider in each of the two conditions, that is the number of samples in condition i is denoted as m_i, $i = \{1, 2\}$. In this thesis we only consider equally sized conditions, that is $m_1 = m_2$. Generally speaking we prefer to have many samples in statistical studies, as we have more information together and obtain better estimates. However, since sequencing is expensive we cannot always expect to have many samples in an experiment. Hence we have restricted ourselves to two different values on $m = m_1 + m_2$, which is $3 + 3$ and $10 + 10$. This should cover both a quite restricted case $(3 + 3)$ and a more generous case $(10 + 10)$, yet still reasonable situations.

Effect size The effect size defines how large the expected downsample count should be when adding the effects in our simulations. The expected count post downsampling, by properties of Binomial distribution, is

$$\mathbb{E}[\hat{Y}_{ij}] = pY_{ij},$$

where p is one over the effect size. This implies that we may expect that an effect size of 10 would reduce the feature count to 1/10 of the original value. Reasonable effect sizes is of course dependent on which data set you are considering, but we will consider effect sizes of 3 and 5. The interpretation of effect size in the model in section 2.1 is that the logarithm of the effect size is estimated by the β in the predictor.

Balance The balance parameter describes how the downsampling of features is distributed between the conditions, as described in section 3.1.1. We will denote the balance by a percentage, describing how many effects added to one of the condition, and assuming that the remaining effects are added to the other condition. The percentage included in this thesis is 50%, 70% and 100%, thus describing a balanced case, a small unbalance and a full unbalance.

Fraction of affected features The last experimental parameter is the fraction of affected features. As the name suggests, it specifies how many features we are adding effects to as a fraction out of the total number of features in the data set. As previously mentioned the features with added effects are what we want to identify in our simulations. In this thesis we include 5%, 10%, 20% and 30%, which represents situations with small differences between conditions (5%) up to situations where the difference between conditions are very notable (30%).

3.2.2 Performance measures

To evaluate the performance we consider the Benjamini-Hochberg adjusted p-values obtained from the F-test applied to the overdispersed Poisson model, as explained in section 2.1. By comparing the adjusted p-values against the significance level α, we consider the test outcome to be positive if $p < \alpha$, and otherwise negative. A positive outcome implies that our model considers the current feature to be differentially abundant, but we must also distinguish between true and false positives, since a positive may correspond to either a DAF or not a DAF.

These binary relations imply that we have four possible outcomes in each test, namely true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN). For comparisons we usually measure the values as rates, i.e. true postive rate (TPR) and false positive rate (FPR) defined as

$$TPR = \frac{TP}{TP + FN}, \quad FPR = \frac{FP}{FP + TN}.$$

TPR is sometimes referred to as sensitivity, and relates to the probability of a positive classification being correct, or simply how many of the actual DAFs are identified. The FPR estimates the probability of falsely rejecting the null hypothesis, i.e. how many non-DAFs are reported to be DAFs by our model.

It should be noted that there are other measures that could be interesting to study, for instance precision (fraction of positives that were true), but in this thesis we will only consider TPR and FP.

Ideally we want a large TPR and a small FPR, which would imply a well-performing classifier. However, the obtained results from the classifiers are often far from ideal. For instance, classifying everything as negative would yield $(TPR, FPR) = (0, 0)$, and similarly, everything as positive implies $(TPR, FPR) = (1, 1)$. Furthermore, as we are evaluating performance of methods by comparing both TPR and FPR, we must consider under which conditions on both these quantities a method should be deemed superior.

We suggest as an alternative approach to compute the TPR at a fixed FPR. That is, instead of comparing the p-values to some significance level α, we consider all

p-values in increasing order and consider each one to be significant (positive) until reaching a certain FPR. The implications of this will be more thoroughly discussed in chapter 5.

4
Results

The results presented in this thesis focuses on how the experimental parameters described in section 3.2.1 influence the different normalization methods. First, we present a few general results for the purpose of assessing how the normalization methods influence the data at a large scale. Next, we focus on the main part of the thesis, which is understanding how the experimental parameters influence the performance of methods, as well as evaluating situations where the methods may be particularly well-performing or poorly performing.

The presented results were obtained using data from two data sets, in this thesis denoted Qin [25] and TARA [26]. The data was pre-processed by removing features with very low abundance, more specifically by only including features with more than 75% non-zero counts and a mean abundance greater than 3. After pre-processing the Qin data set contained the abundance of 3592 features and 145 samples divided into two conditions connected to individuals with and without stage I diabetes. Meanwhile, TARA contained 4787 features and 76 samples divided into two conditions related to the surface layer and deep chlorophyll maximum layer of ocean water.

4.1 Performance evaluation on raw data

First, we compare the performance of the methods on raw data by measuring the coefficients of variation c_v across all samples in one condition. The coefficient of variation is a standardized measure of the variance in the data, computed as

$$c_v = \frac{\sigma}{\mu},$$

where σ is the standard deviation and μ is the mean. The motivation behind normalizing is to remove unwanted variation in the data, and hence we expect that the variance in the data should decrease after being normalized.

Figure 4.1 shows c_v for non-normalized data and for all nine normalization methods (see section 2.2). As it can be seen, all methods reduce the variance of c_v. However, while the median c_v is approximately equal across all methods, we see that the variance of c_v is smallest for CSS, followed closely by normalizing by Med and then Quantile.

Second, we evaluate pairwise agreement of DAFs identified after normalization by different methods. Note that we do not take into consideration if the identified DAF is truly a DAF or not. Figure 4.2 and figure 4.3 are heat maps of the number

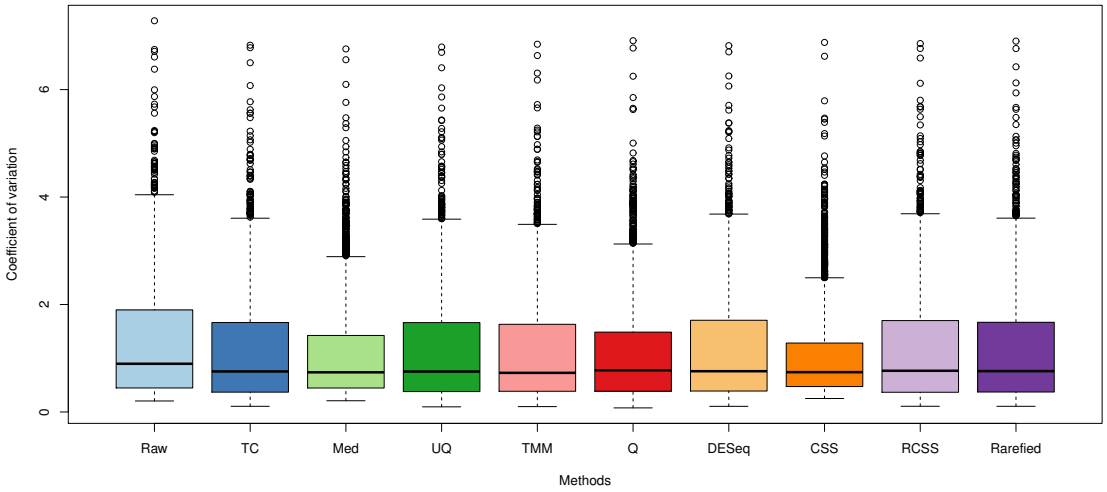


Figure 4.1: The coefficients of variation for each normalization method, computed across samples in one condition for each feature in the Qin data set.

of DAFs found by the oGLM after normalization of the data using each one of the studied methods, presented as symmetric matrices M where each entry represents the number of features that were considered differentially abundant by two methods separately. That is, a diagonal element M_{ii} represents the number of DAFs found by method i, while any other element M_{ij} is a count of the number of features that were DA in both method i and method j.

Considering the diagonals of the two figures, we note that CSS leads to a much larger amount of significant features as compared to the other methods. Med also reports a somewhat larger number of significant features, although not as extreme as CSS. We also note that Quantile seems to be more conservative as compared to the other methods. The remaining methods seem to be approximately equal in terms of reported DAFs.

If we consider the remaining elements, we note that TC, UQ, TMM, DESeq and RCSS have rather small differences in their resulting DAFs, as the number of DAFs in their intersections usually are reasonably close to the number of DAFs reported by either method. For instance, in figure 4.2 TC reports 1791 significant features and TMM reports 1875 significant features, while the number of DAFs common for both methods is 1651. Even closer connected are DESeq and TMM, where DESeq reports 1868 and their intersection is 1850. At the same time, CSS and Med seems to share many DAFs as well. Similar results are visible in figure 4.3, but we should note that the exceptionally high amount of DAFs reported under CSS, as compared to the other methods, may potentially be misguiding. At the same time we may observe that CSS is quite distant from most methods in terms of common DAFs, considering how many DAFs the method actually reports. For instance, even though 1299 DAFs are reported under CSS normalization, only 159 out of the 221 DAFs reported under TC are also reported under CSS, which may indicate that CSS generates more FP than the other methods.

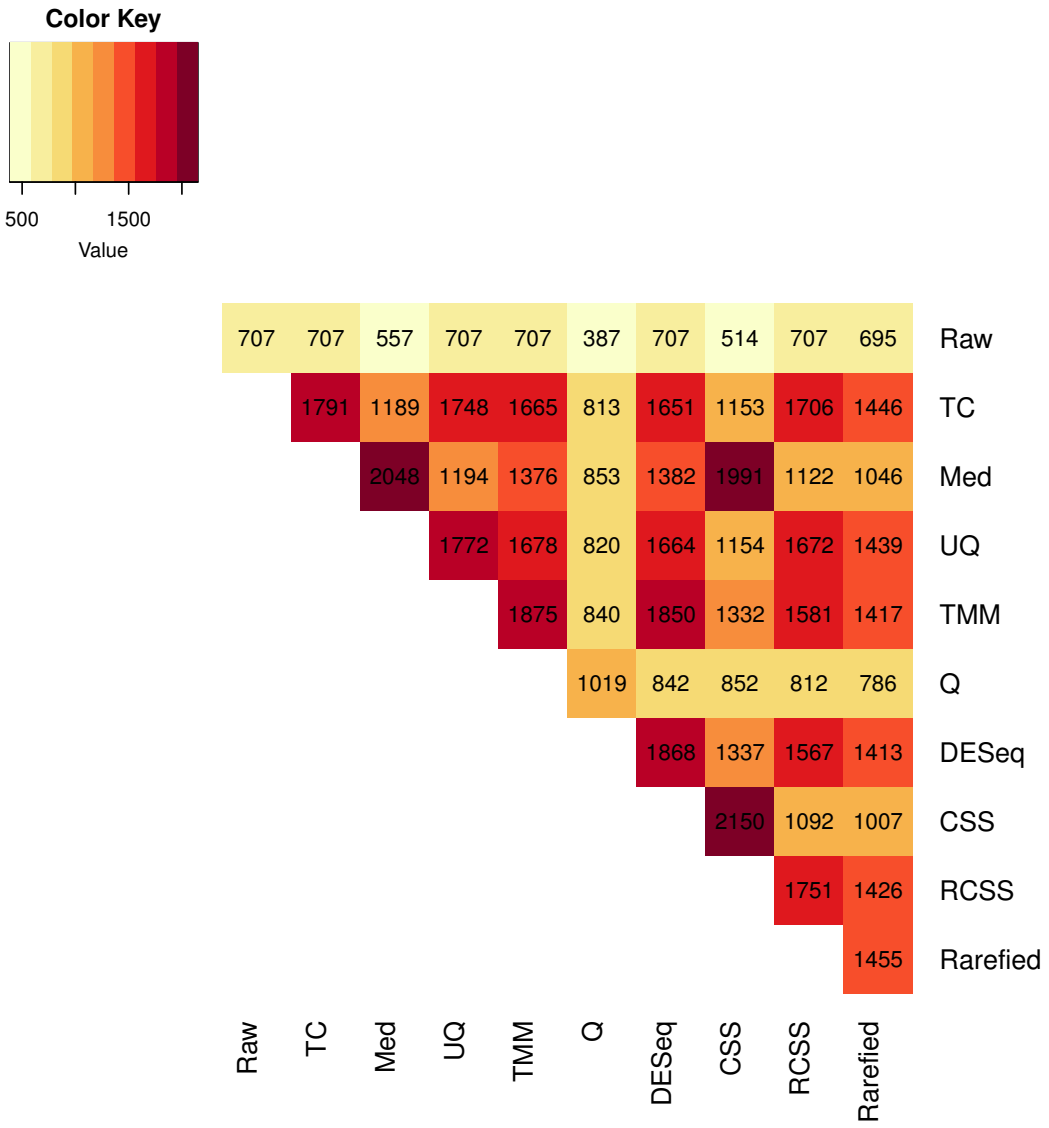


Figure 4.2: The number of features reported significantly differentially abundant by the oGLM, when using different normalization methods. Each entry shows the number of features identified as differently abundant by the two methods in the corresponding row and column. The data set considered here was TARA, which contains 4787 features and $(41 + 35)$ samples.

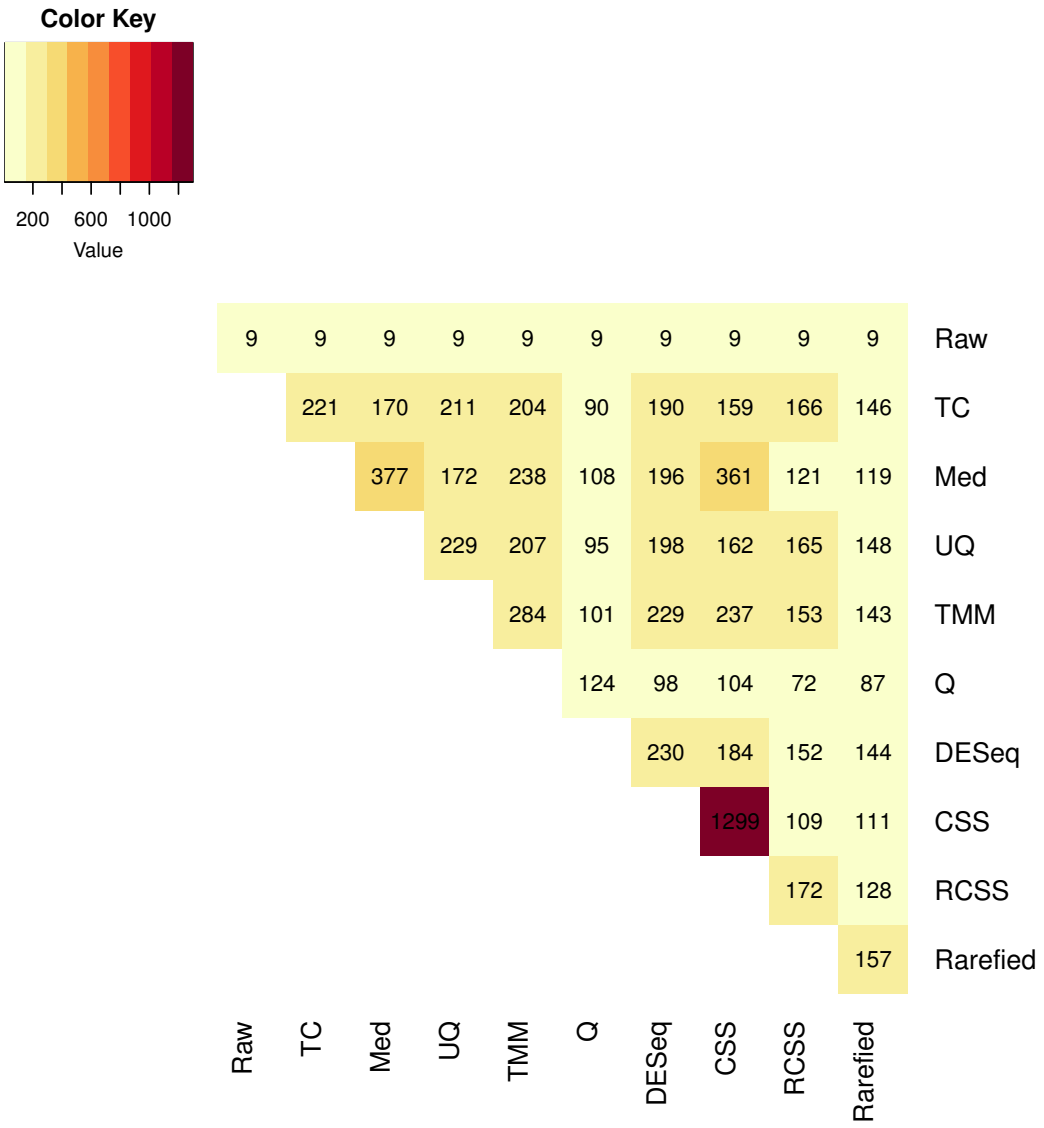


Figure 4.3: The number of features reported significant by the oGLM, when using different normalization methods. Each element shows the number of features identified as differently abundant by the two methods in the corresponding row and column. The data set considered here was Qin, which contains 3592 features and $(71 + 74)$ samples.

4.2 Performance evaluation on data with artificial effects

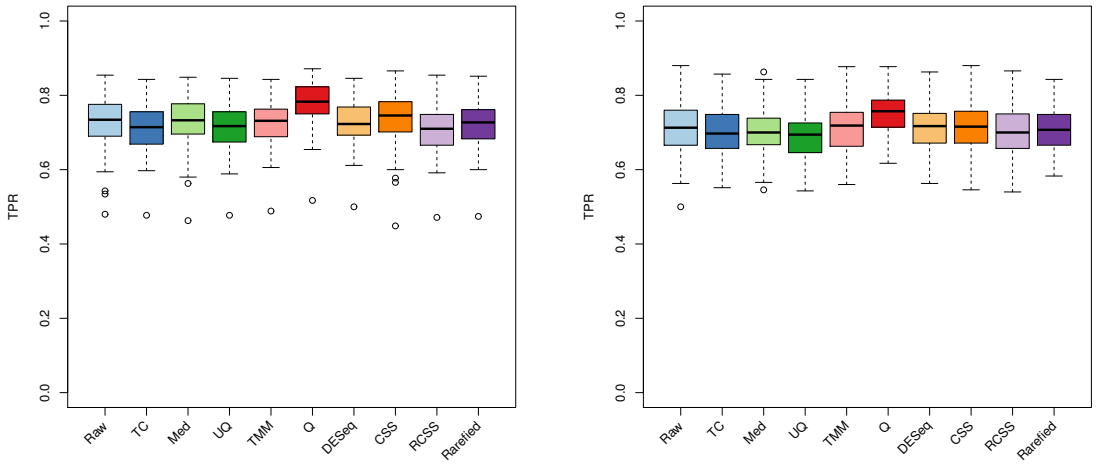
Moving on from the real data analysis, we now treat results obtained from bootstrapping the data from one condition (see section 3.2 for details). In this thesis we used the stage I non-diabetic samples in the Qin data set as described in the beginning of this chapter. Furthmore, for each resampled data set we randomly picked 3500 features to be included after excluding any feature with all counts equal to zero.

Effects were added according to the experimental parameters: condition size, effect size, proportion of affected features and balance presented in details in 3.2.1. Here, we will present results obtained by varying one of these parameters at a time while fixing the others. All results presented here are based on 100 bootstrapping iterations, which should be a sufficiently large amount of iterations in order to consider that any pre-existent DAFs in the data will in average not contribute to our results, hence we may assume that there are no DAFs in the data but the artificial DAFs intentionally added by downsampling a specific number of features in one condition.

4.2.1 Distribution of effects between conditions

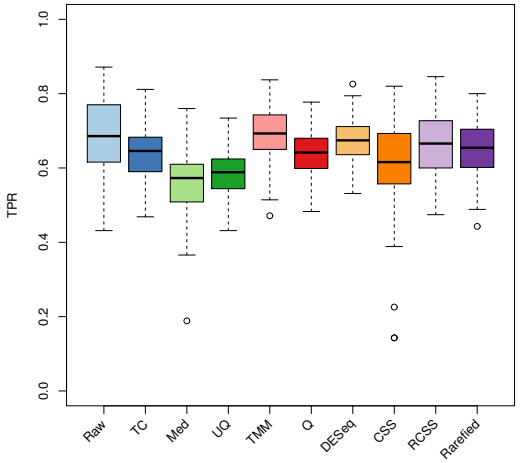
The distribution of effects between conditions determines the number of artificial effects added to either condition, i.e. if the distribution is said to be 70%, then 70% of the effects are added to one condition and the remaining fraction to the other condition. Since the samples for each condition are randomly chosen in each bootstrap iteration, it is sufficient to consider adding 50% to 100% of the effects to one condition, as the results are expected to be symmetrical. That is, adding 30% of the effects to condition 1 should be equivalent to adding 30% of the effects to condition 2 in terms of observed performance.

Figure 4.4 shows the true positive rate (TPR) with a fixed false positive rate (FPR) of 0.01 for each method while altering the distribution of effects between artificial conditions, i.e. having the distribution set to 50%, 70% and 100%, with the remaining experimental parameters fixed to $10 + 10$ for the condition size, 5 for the effect size and effects added to 10% of the features. The figure shows that the performance decreases for all methods as the effects are added more unbalanced between the artificial conditions. In particular, while CSS and Med performs well for the balanced case, both methods perform poorly under unbalanced effects. Quantile is clearly the superior method under balanced effects and small unbalance, while still decent under full unbalance. TMM and DESeq are only marginally affected by the unbalance.



(a) Distribution of effects: 50%

(b) Distribution of effects: 70%



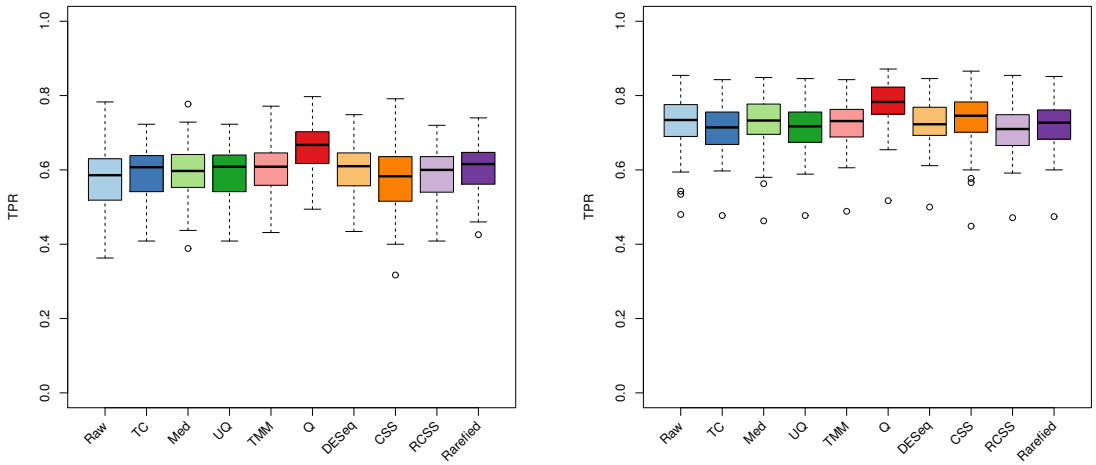
(c) Distribution of effects: 100%

Figure 4.4: True positive rate at fixed false positive rate $FPR = 0.01$. The experiment was performed for $10 + 10$ samples, with an effect size of 5 and with 10% of the 3500 features affected. Each panel shows a boxplot of the TPR in each of 100 bootstrap iterations, when the affected features are added 50% in each condition (a), 70% and 30% in each condition (b), or all at the same condition (c).

4.2.2 Effect size

The effect size represents how large the artificial effects are. The effect size determines the probability p in our binomial downsampling, as p is equal to one over the effect size. This implies that the effect size corresponds to how large the relative difference in abundance is expected to be.

In figure 4.5 the TPR at fixed FPR of 0.01, with an effect size of 3 and 5, are shown. The fraction of affected features were 10%, the distribution of affects between conditions were 50% and the condition size were $10 + 10$. The figure shows that all methods perform better for the larger effect size 5. In both cases Quantile outperforms all other methods. For the small effect size, all the remaining methods perform equally or better than not normalizing, however under large effect size we note that TC, UQ, RCSS and Rarefying have a lower median TPR than no normalization.



(a) Effect size of 3.

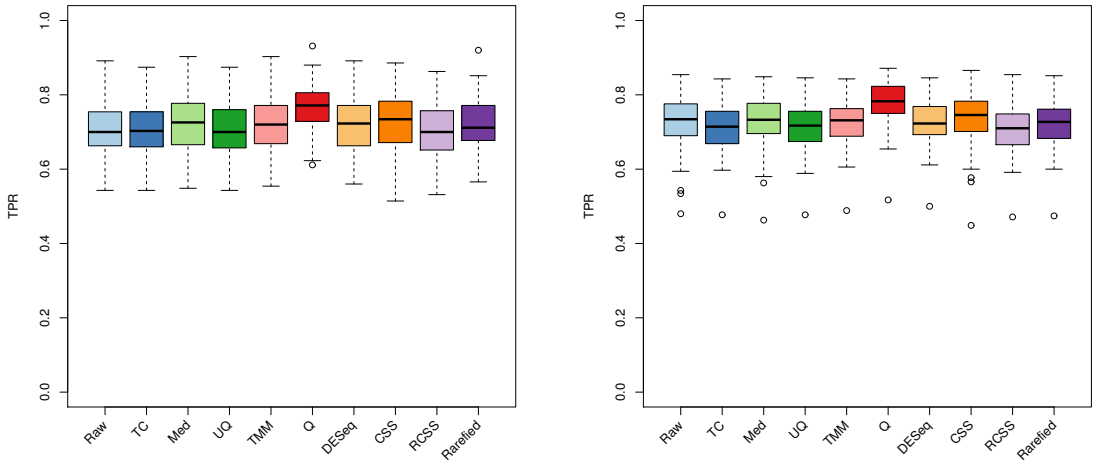
(b) Effect size of 5.

Figure 4.5: True positive rate at fixed false positive rate $FPR = 0.01$. The experiment was performed for $10 + 10$ samples, with effects distributed equally between conditions and with 10% of the 3500 features affected in both panels. We used effect size of 3 (a) and 5 (b). Each panel shows a boxplot of the TPR for each of 100 bootstrap iterations.

4.2.3 Fraction of affected features

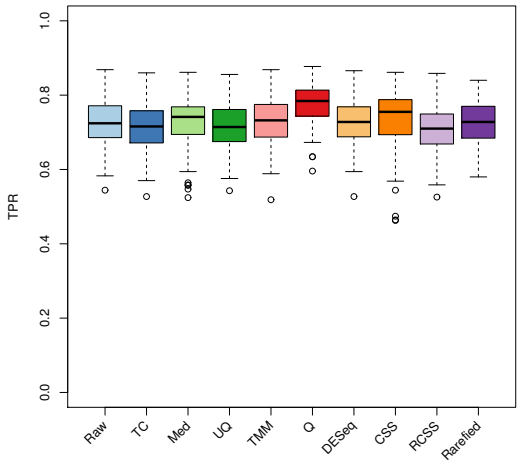
The fraction of affect features specifies the fraction of features with added effects, i.e. artificial DAFs, as a fraction of the total number of features, here 3500.

Figure 4.6 shows the TPR at fixed FPR under a varied fraction of affected features, namely 5%, 10%, 20% and 30%. The distribution of effects between conditions was 50%, while the condition size was $10 + 10$ and the effect size was 5. The figure shows that despite increasing the fraction from 5% to 30%, there seems to be no noteworthy difference in the performance, considering both comparisons between methods and in the obtained TPR.

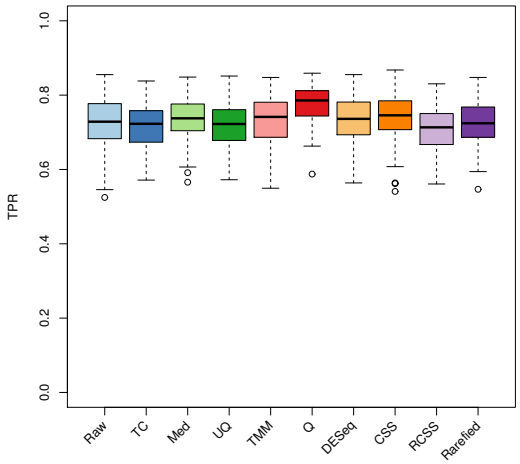


(a) 5% affected features.

(b) 10% affected features.



(c) 20% affected features.



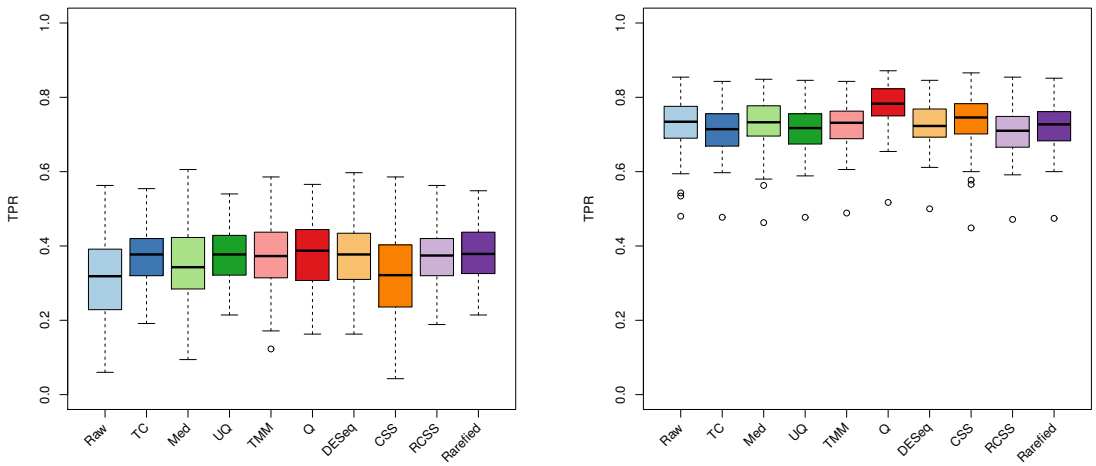
(d) 30% affected features.

Figure 4.6: True positive rate at fixed false positive rate $FPR = 0.01$. The experiment was performed for $10 + 10$ samples, with an effect size of 5 and with effects distributed equally between conditions. The fraction of affected features considered were 0.05 (a), 0.1 (b), 0.2 (c) and 0.3 (d). Each panel shows a boxplot of the TPR for each of 100 bootstrap iterations.

4.2.4 Condition size

The condition size specifies the number of samples in each condition, which simply speaking implies how much information we have about each condition. We consider two options, namely $3 + 3$ and $10 + 10$, i.e. 3 or 10 samples in each artificial condition.

The TPR at fixed FPR of 0.01 with varied condition size is shown in figure 4.7. The distribution of effects between conditions was 50%, the fraction of affected features was 10% and the effect size was 5. We note that there are large differences in performance between condition sizes for all methods. We also note that for small condition sizes most methods are superior to not normalizing, with the exception of CSS that performs similarly to not normalizing. Furthermore, all of the remaining methods except Med performs approximately equally well. However, as the condition size is increased to $10 + 10$, we note that TC, UQ and RCSS are surpassed by not normalizing, and also the Quantile is now superior to all other methods.



(a) $3 + 3$ samples.

(b) $10 + 10$ samples.

Figure 4.7: True positive rate at fixed false positive rate $FPR = 0.01$ for each normalization method. The experiment was performed with an effect size of 5, equally distributed effects between conditions and with 10% of the 3500 features affected. Each panel shows a boxplot of the TPR for each of 100 bootstrap iterations, and condition size $3 + 3$ (a) and $10 + 10$ (b).

4.3 Simulation of biological situations

The previous section covered how each method behaved when each experimental parameter was varied under the assumption that the remaining parameters were fixed to a relatively easy situation. However it is possible that the performance is affected in different ways when experimental parameters vary simultaneously, for instance by amplifying or diluting the influence on performance of each individual experimental parameter.

Figure 4.8 shows the obtained TPR (left panel) and FPR (right panel) when adjusting the p-values from oGLM according to the Benjamini-Hochberg procedure with a false discovery rate of 0.05. The figure shows the results from a situation with balanced distribution of artificially created DAFs (50%), i.e. features where effects are added by downsampling the counts, but with small condition size $(3 + 3)$, large fraction of affected features (30%) and large effect size (5). We note that under these situations, normalization in general increases the reported number of true positives, while still keeping a low FPR. We also see that most methods perform equally well, with the exception of Med and CSS, for which the TPR is notably lower.

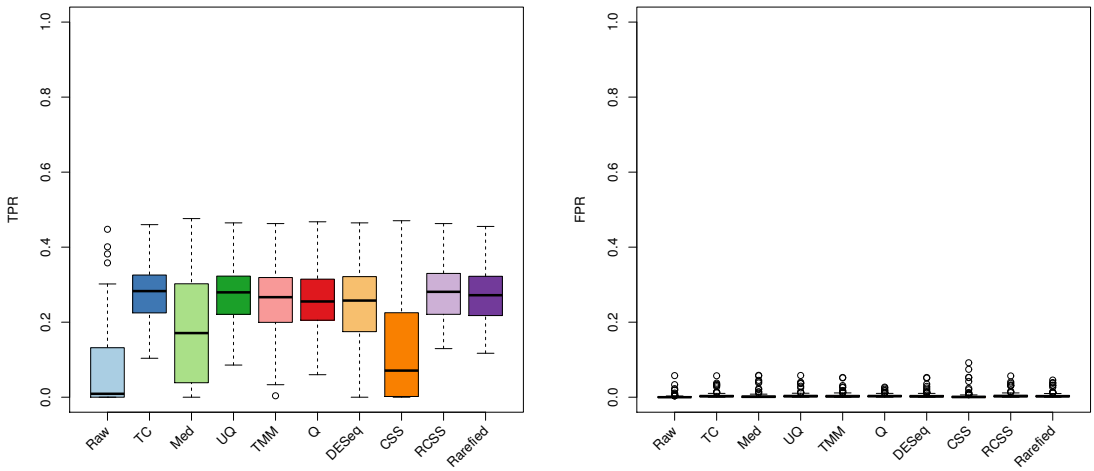


Figure 4.8: True positive rate and false positive rate of different normalization methods, obtained from p-values adjusted according to Benjamini-Hochberg under a false discovery rate of 0.05. The effect size was 5, the condition size was $(3 + 3)$, the fraction of affected features was 30% and the distribution of effects between conditions was 50%.

If we instead study larger groups, i.e. $10 + 10$, together with a relatively small effect size of 3, while keeping the same fraction of affected features to and the distribution of effects between conditions as in the previous figure, that is 30% and 50% respectively, we obtain the results shown in figure 4.9. The most notable difference is that the TPR of all methods is higher in comparison to the situation with smaller condition sizes and larger effect size, as seen in figure 4.8.

Furthermore, the relative difference between methods are clearly smaller with a larger condition sizes. If we consider the smaller condition size, we see that not only does CSS and Med perform worse according to the median of the TPR, but the boxes are also notably larger suggesting that the expected performance is less reliable than for other methods. The difference in box sizes is however reduced for larger condition size, again suggesting that the number of samples in a condition is very important for both normalizing, but also for the statistical test.

Quantile performs best of the methods, with the highest TPR and lower FPR. The FPR is close to equal for all methods, and the median of the FPR is actually less than a percentage point for all methods. Hence we may conclude that all methods perform well under these conditions, but with Quantile being a superior choice of method. We do however know from the results in section 4.2.1 that balance is an important factor to consider, as we saw decreasing performance in terms of TPR at fixed FPR for most methods when shifting the distribution of effects from balanced to unbalanced situations. The decrease in performance was very notable even for moderate choices of the other experimental parameters.

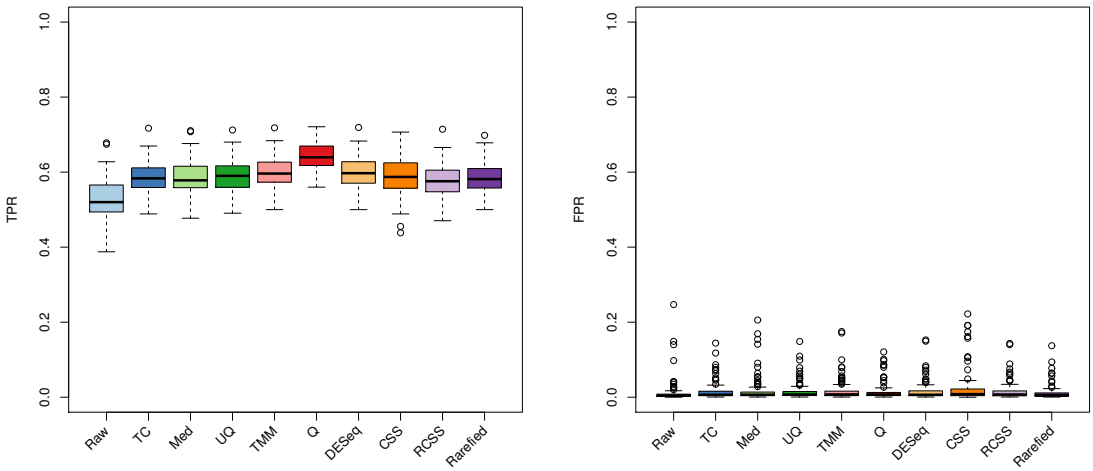


Figure 4.9: True positive rate and false positive rate of different normalization methods, obtained from p-values adjusted according to Benjamini-Hochberg under a false discovery rate of 0.05. The effect size was 3, the condition size was $(10 + 10)$, the fraction of affected features was 30% and the distribution of effects between conditions was 50%.

In figure 4.10 some unbalance in terms of the distribution of artificial DAFs is introduced to the data sets, namely we added 70% of the effects to one condition, and the remaining 30% to the other condition. The other experimental parameters were unchanged from figure 4.9, i.e. the condition size was $10 + 10$, the effect size was 3 and the fraction of affected features was 30%. Comparing the TPR between the two figures we note that adding a small unbalance of effects between conditions does not affect the overall performance of the methods to any large extent, with the exception of CSS and Med for which the TPR is decreased somewhat. However, even with a rather small unbalance, we begin to see an increase in FPR for all normalization methods. While all methods exceed 1 percentage point in median of the FPR, the UQ for instance approach 4 percentage points.

In figure 4.11 we increased the effect size to 5, while maintaining condition size as $3 + 3$, the fraction of affected features as 30% and the distribution of effects between conditions as 70%. In this figure we observe an increase in TPR, but also an even larger increase in FPR. We also note that especially Med and UQ, and to some extent Quantile and CSS, are sensitive to this less ideal situation, generating more FP than the other methods. At the same time, TMM seems to be marginally affected by the increased effect size. The performance of DESeq, RCSS, Rarefying and TC are still decent but with a bit lower TPR than the other methods. Not normalizing has the lowest TPR, and also a very low FPR.

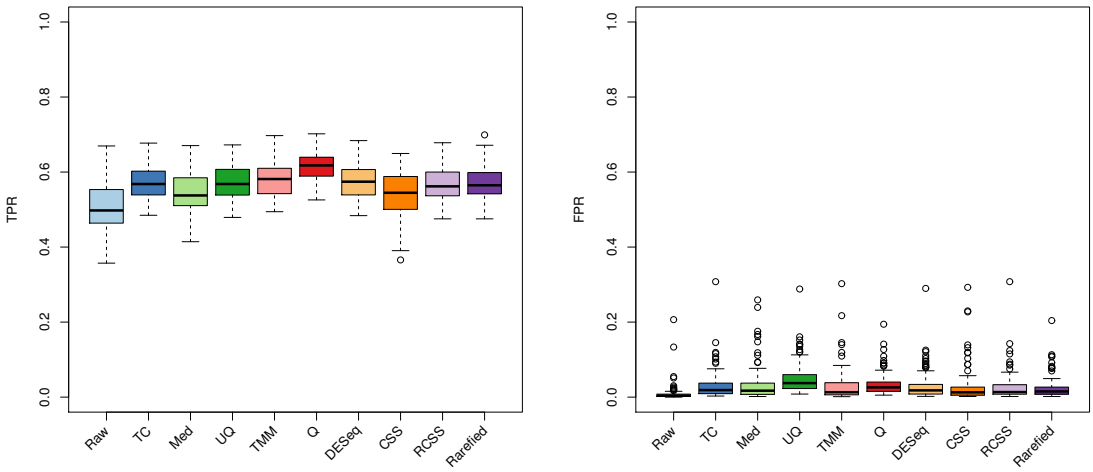


Figure 4.10: True positive rate and false positive rate of different normalization methods, obtained from p-values adjusted according to Benjamini-Hochberg under a false discovery rate of 0.05. The effect size was 3, the condition size was $(3 + 3)$, the fraction of affected features was 30% and the distribution of effects between conditions was 70% and 30%.

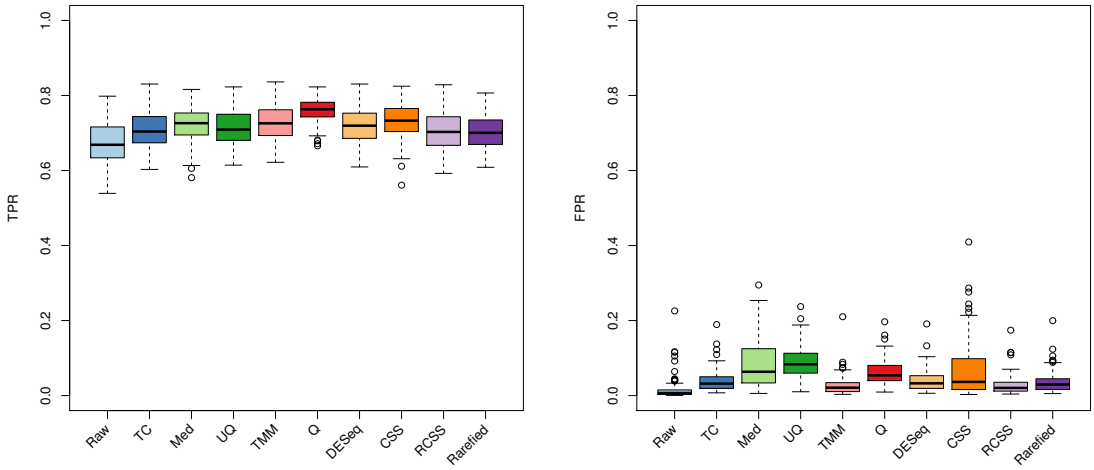


Figure 4.11: True positive rate and false positive rate of different normalization methods, obtained from p-values adjusted according to Benjamini-Hochberg under a false discovery rate of 0.05. The effect size was 5, the condition size was $(3 + 3)$, the fraction of affected features was 30% and the distribution of effects between conditions was 70% and 30%.

If we make the distribution of effects between conditions even more ill-conditioned, that is by only adding effects to one condition, we obtain the results seen in figure 4.12. The condition size was $3 + 3$, the effect size 5 and the proportion of affected features was 30%. The figure shows that the FPR is $> 10\%$ for all normalization methods, and most notably we have that Med, UQ, Quantile and CSS are close to 40% in FPR. TMM and RCSS are the best performing normalization method under these conditions, with a high TPR combined with the lowest FPR. While DESeq has been well-behaving under less extreme conditions, it is clear that this method cannot correctly handle the full unbalance.

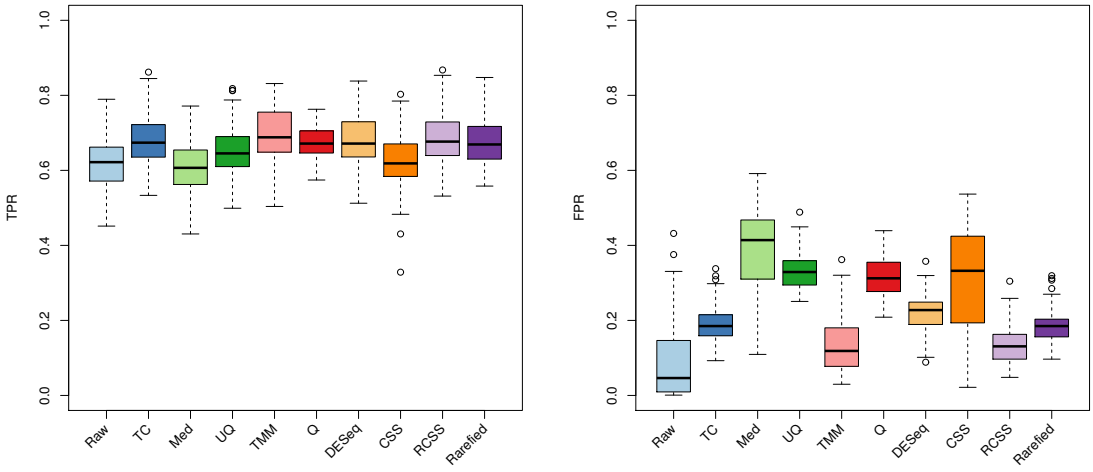


Figure 4.12: True positive rate and false positive rate of different normalization methods, obtained from p-values adjusted according to Benjamini-Hochberg under a false discovery rate of 0.05. The effect size was 5, the condition size was $(3 + 3)$, the fraction of affected features was 30% and with all effects added to only one of the two conditions.

5
Discussion

This thesis considers nine different normalization methods and how well they perform on metagenomic data. The methods studied are total count (TC), median (Med), upper quartile (UQ), TMM, quantile, DESeq, cumulative sum scaling (CSS), reversed cumulative sum scaling (RCSS) and rarefying, and each one of them are described in detail in section 2.2. Overall they normalize the data by computing a sample-wise normalization constant that scales the counts into rates, or by adjusting each count by some assumption, for instance by equalizing each quantile of the data (Quantile). Our main purpose is to understand how they affect the downstream analysis when identifying differentially abundant features (DAFs), that is features for which the abundance is significantly different between conditions. The statistical test used for identifying DAFs in this thesis is an overdispersed Poisson model presented in section 2.1, and our results are based on how this statistical test performs when different normalization methods are applied. Both real data as well as semi-simulated data are considered for the results, where the latter is based on resampling the original data set and adding artificial DAFs (see section 3.2 for further reading on simulations).

Ultimately, we are interested in how the methods perform on real metagenomic data. However, due to the fact that the data is noisy and we do not know beforehand which features that are true DAFs (for a longer discussion on dealing with real data, see chapter 3), correctness of the identified DAFs cannot be verified. Nevertheless, for the sake of comparison between the normalization methods and their power, it is still possible to make some small observations regarding differences between normalization methods on real data. The first of these observations is that normalizing does decrease the intra-condition variance, which was measured by observing a decrease in the coefficients of variation before and after normalizing the data using each one of the nine methods studied here (figure 4.1).

However it is generally hard to determine whether all this reduction of variance is wanted. We see that especially CSS seems to have a trend of very large reduction of intra-condition variance. Furthermore, when considering real data the number of DAFs reported by CSS was notably higher than for most other methods, while a large number of DAFs reported by CSS was not found by the other methods, as was observed when studying the reported DAFs under different normalization methods (figure 4.2). While it is of course possible that CSS is correct and the other methods are erroneous, it is more likely that CSS manages to reduce variance to such large extent that valid information in the data is lost and a lot of misclassifications happen. Similar observations hold for Med, and are strengthened by noting that most other methods manage to identify all the DAFs found when not using any normalization

method.

Another important observation is that we in this thesis only include one statistical test for identifying DAFs. While we expect that the power of normalization methods observed when using oGLM should be consistent when used together with other statistical tests, we of course add assumptions to the data. This must therefore be taken into consideration when studying the results. For further analyses on different statistical tests, see Jonsson 2016. Furthermore, when applying normalization methods and oGLM on two different data sets, the number of reported DAFs differed considerably. For instance, on the TARA data set, the absolute differences were generally smaller between CSS and the others, while for Qin2012, the difference was very large (figure 4.2 and figure 4.3). However, in order to generalize the data sets, we used variations of different experimental parameters, each thoroughly explained in section 3.2, in order to simulate different situations.

5.1 Simulated situations

As previously discussed, as we lack the possibility of using real data for the full analysis of normalization methods, we instead simulated different situations specified by the experimental parameters, i.e. condition size, effect size, fraction of affected features and distribution of features between conditions. The purpose of this study was to reflect different possible real life scenarios, while still using real data to reflect the large variability and noise present in it. In section 4.3 we presented some situations that were interesting when considering the normalization methods, and we will now discuss the interpretations and implications of the results.

In the first simulated situation studied we considered a situation where the added effects, i.e. the DAFs, were equally distributed between the two conditions (50%), and we had a large fraction of affected features (30%). The effect sizes, i.e. the difference between the conditions, were large (5) and we considered few samples in each condition ($3 + 3$) (figure 4.8). From a biological perspective, this could represent as a situation where we study two environments, considering the whole set of features (i.e. genes or genetic functions) sequenced from the samples. The balanced distribution of DAFs suggests that if one of the environments is missing some unique function, it can be compensated through some other feature. Furthermore, the large fraction of affected features and the large effect size suggest that the environments are very different, and lastly, few samples from each environment is available for analysis. The conclusion from the results is that identifying DAFs in this situation is hard, even though the effect sizes suggests that the DAFs should be rather easy to identify. The low TPR is most likely due to limitations in the statistical test, as we have a very limited data set in terms of number of samples. We also have a large number of DAFs in this situation, which should violate several assumptions in several methods, for instance TMM and DESeq that assume that few features are differentially abundant. About half of the normalization methods also use information across samples when normalizing, which suggests that they should be less reliable in situations with few samples. In situations like this, the suggestion would be to avoid using Med or CSS, as they behave notably worse than the other methods, and if possible increase the number of samples in order to increase the general

power. Normalizing is however clearly important and should not be excluded from the chain of processing.

In the next situation considered we had a larger number of samples in each condition ($10 + 10$), while instead a smaller effect size (3). Meanwhile, the fraction of affected features was large (30%) and the effects were distributed equally (50%). In other words, we have a situation similar to the previous case, but with more available data, i.e. samples, while the differences between the two conditions are a bit smaller in terms of the effect size, but still having a relatively large number of DAFs. In other words, we are for instance studying two environments where we have taken several samples from each environment, there are many differences between the two environments, although they are small in size with an effect size of 3, which makes them hard to be identified. The results showed a large increase in power for all methods (figure 4.9), as compared to the previously described situation (figure 4.8). Even though the decreased effect size makes it harder to differentiate random variations from DAFs, a large number of samples does increase the guaranteed power, as the TPR is much higher than the previous situation. Under these circumstances, the recommendation would be to use Quantile as normalization method, which guarantees a very high power as compared to all other normalization methods.

In the two previous situations we had an equal distribution of effects between the two conditions. If we consider the same situation as the previous one, that is a large number of samples in each condition ($10 + 10$), a large number of DAFs (30%) and small effect size (3), but with an unbalanced distribution of effects between the conditions (70%). This could represent a biological study where we compare the gene abundance looking for some specific function, that is a target sequencing situation where we study the ability of two quite different environments with regard to some function. In such case we may expect one environment to have loss of this function while the other retains it, i.e. that the function is generally less abundant in one of the environment, translating to a shift in the distribution of effects. The results showed a clear increase of FPR (figure 4.10) as compared to the previous case, i.e. a similar situation with the exception of the distribution of effects instead being balanced (figure 4.9). The difference in TPR is largely negligble. Under this situation, normalization is still considered to be important. Note, however, that even though the FPR is larger than in balanced situation, it is still quite small. The high TPR of Quantile suggests that this method is superior in this situation as well, where the effects size are small (3).

If we increase the effect size we again arrive to a situation where the samples are very different between the two conditions. More precise, we have a situation with large effects (5), large fraction of affected features (30%), large number of samples ($10+10$), and with a small unbalance in the distribution of affected features between conditions (70%). This situation is representative for, much like the previous situation, a biological case where we are studying a biological function that is expected to have a higher abundance in one condition relative to the other condition, but with the difference in abundance of this function is even larger than in the previous situation. This corresponds to biological cases where the environments are very different in several ways, i.e. there are many features with different abundance, the abundance of this feature is very different and the number of DAFs are quite differ-

ent between the two environments. We found that we obtain a significantly higher TPR in general in comparison to the similar situation but with smaller effects (figure 4.10 and figure 4.11). Since the effect size should make the the artificial DAFs more easily identified, this behaviour is expected. However, the increase in FPR suggests that the number of significant features, i.e. positives, reported by our statistical test has increased quite notably. That is, the difficulty to distinguish real DAFs from ordinary differences has increased, especially when considering normalizing in comparison to not normalizing. As the normalization methods ideally would reduce the general variation between all samples, while maintaining the different abundance for DAFs, we can conclude that this situation starts to show drawbacks of normalizing over not normalizing. Still, the high TPR of Quantile is still very notable which makes it a superior method if we can accept the larger FPR when comparing it to TMM for instance.

Lastly, we considered a situation with large effects (5), few samples (3 + 3), large number of affected features (30%) and where all effects were added to one condition, i.e. a fully unbalanced distribution of affected features (100%). This situation corresponds to the study of two very different environments, targeting some function that is more or less absent in one condition. Furthermore, there are a lot of features corresponding to this function, and the situation should be considered hard as several methods rely on that the distribution is equal for all samples, but when a large number of effects are added to one condition, the distribution will likely be quite different. In short, this situation is very extreme, but interesting as it also shows extreme behaviour of the normalization methods.

We found that normalization is in most cases increasing the TPR, with the exception of Med and CSS (figure 4.12). However, as the FPR increases to very high levels, which for instance occurs when all effects are added to one condition, the suggestion would be to either use TMM, which is clearly less sensitive to this type of error, or maybe consider not normalizing at all. In fact, the TMM does increase both TPR and FPR, and the trade-off between the two should be considered from case to case, i.e. to what extent false positives can be ignored. It should be noted that these results are based on semi-simulated data, and the assumptions of the added effects do of course influence the results, especially in this case. As explained in section 3.1.1, if we add effects to one condition, we also downsample all features in the other condition. If, like in this situation, we have many DAFs, large effects and all effects in the same condition, we will in our simulations increase the difference in abundance of non-DAFs as well. Furthermore, the downsampling applies to all features in the condition with no added effects, implying that the observed difference in abundance decreases in this situation. In this case, adding effects to 30% of the features in one condition with effect size 5, would downsample all features in the second condition by an effect size of approximately 1.3. While this of course is not a large effect compared to the effect size 5 used for actual artificial DAFs, it is likely that the combination of extreme values of the effect size and the fraction of affected features amplifies the FPR found in unbalanced situations.

5.2 Similarities between methods and their assumptions

The normalization methods included in this thesis are all based on some assumptions. In some cases these assumptions are the same for two or more methods, and these assumptions often boil down to similarities between the normalization methods. We briefly mentioned some similarities in section 2.2, however with the presented results in chapter 4, we may also observe if the results behave similarly, and possibly even argue for what makes a method good. To begin with, we note that some methods, such as TC, CSS and RCSS, normalize by the total counts in the whole sample or for some subset of the data. We argue that TC and RCSS should perform similarly to each other, as RCSS considers the top 25% of the counts and the high-count features are expected to give a large contribution to TC. Consequently, since CSS normalization is defined as the sum of counts smaller or equal to some quantile, and at most 50% of the counts, it is possible that these counts behave very differently than the top 50% of the counts. We note that none of these methods are particularly well-performing, but TC and RCSS seem to generally perform better than CSS when considering either small condition sizes $(3 + 3)$ as seen in figure 4.7 and figure 4.8, while CSS performs better for larger condition size $(10 + 10)$, as seen in figure 4.9. Even if CSS might be an overall good normalization method, the method relies on estimating a quantile that determines the range of the sum (see section 2.2 for details), it is possible that estimating this quantile is very sensitive to the condition size, and hence the method becomes unreliable when we have few samples. Also, since TC and RCSS have no assumptions on the number of samples in each condition, their observed power is mainly determined on how the statistical test is influenced by number of samples.

Another mentioned similarity was which assumptions on the distribution of each sample are used by each normalization method. While TC and Rarefying essentially assume that the total amount of information in each sample should be equal, most of the other methods make assumptions regarding the distributions of counts. For instance, Med assumes that if we let the median of each sample to be equal, then the normalization is accurate. Meanwhile, Quantile is much stricter as it equalizes each quantile across samples. DESeq, TMM, UQ and CSS also make assumptions on the distribution to some degree. While assumptions on the distribution are most likely reasonable, the implications of each assumption differ a lot. For instance, figure 4.4 showed the TPR at FPR 0.01 when changing the distribution of affected features from balanced to unbalanced, with the remaining experimental parameters fixed. The figure showed a significant decrease in power for most normalization methods, however most notably for Med, UQ, Quantile and CSS. One explanation could be that when the distribution of effects is equal between conditions, the distribution of all counts may also be assumed to be close to equal. However, if all effects are added to only one condition, then we actually expect that only parts of the counts should be distributed equally. For instance, even if the median and upper quartile are robust to outliers, they are expected to be sensitive to heavy downsampling in one condition, as both the median and upper quantile of the counts will be shifted downwards in this condition. Meanwhile, if we downsample equally often in both

conditions, then we expect that the impact on the median and upper quartile of counts behave similarly for both conditions. The reason why for instance TMM is less sensitive to this kind of unbalance is likely because of the method considers a trimmed set of features, and should suffer less when one condition contains all the added effects.

5.3 Performance measures

In this thesis we focused on two related performance measures, TPR at fixed FPR and TPR/FPR at a fix FDR. First, the TPR at FPR = 0.01 makes the comparison of methods easier as it includes both the TPR and FPR in the same measure. It is very informative when studying how changing some experimental parameter influences the results. However, it is not informative in the sense of what the observed power will be in applications. This is due to the fact that the figure only explains the relative order of p-values corresponding to artificial DAFs and non-DAFs. Hence, two normalization methods that are comparable in this measure may have resulted in very different p-values, for instance very small (many positives) or very large (few positives). To take the actual p-values into consideration, we used the second measurement used is TPR and FPR obtained from p-values adjusted to an FDR of 0.05. While these results are more in line with what will be observed in applications, it may in some cases be hard to distinguish which method is preferable. Furthermore, the p-values are of course directly related to the statistical test, thus making them biased and not necessarily representative to other statistical tests for identifying DAFs. Hence, to fully determine how well methods perform both kind of figures should be taken into consideration, and ideally other measurements not included in this thesis as well, for instance FDR and distribution of p-values, but also studies on other data sets.

5.4 Conclusion

In this thesis we have presented and evaluated nine normalization methods that are applicable to metagenomic data. The results presented has shown that normalization is a vital step when identifying differential abundant features in metagenomic data. We simulated different situations by varying condition size, effect size, number of affected features and the distribution of effects between conditions, and studied how each normalization method performed in these situations. It is clear from the results that no normalization is superior at all time, although the Quantile was proven to be a very strong normalization method in most situations. Further, we conclude that TMM and DESeq are reliable in general, as they performed decently even in the more extreme situations. We also found that not normalizing may be a viable approach when considering situations when all DAFs have higher abundance in the same condition.

When considering further development of normalization methods, we conclude that assumptions regarding the distribution of counts are generally well-performing. A possibility for improvements would be to study methods for identifying features

that are outliers to the distribution in order to exclude them, as they most likely have a negative influence on the normalization. Several methods already consider this problem, for instance CSS, TMM and the non-included tool RAIDA, however there is most likely room for further improvements in how these features should be selected.

Bibliography

[1] Mincheol Kim, Ki-Hyun Lee, Seok-Whan Yoon, Bong-Soo Kim, Jongsik Chun, and Hana Yi. Analytical tools and databases for metagenomics in the next-generation sequencing era. *Genomics & informatics*, 11(3):102–113, 2013.

[2] Torsten Thomas, Jack Gilbert, and Folker Meyer. Metagenomics-a guide from sampling to data analysis. *Microbial informatics and experimentation*, 2(1):1, 2012.

[3] Michael B Sohn, Ruofei Du, and Lingling An. A robust approach for identifying differentially abundant features in metagenomic samples. *Bioinformatics*, page btv165, 2015.

[4] James Robert White, Niranjan Nagarajan, and Mihai Pop. Statistical methods for detecting differentially abundant features in clinical metagenomic samples. *PLoS Comput Biol*, 5(4):e1000352, 2009.

[5] Thomas J Sharpton. An introduction to the analysis of shotgun metagenomic data. *Frontiers in plant science*, 5:209, 2014.

[6] Stephen Nayfach and Katherine S Pollard. Toward accurate and quantitative comparative metagenomics. *Cell*, 166(5):1103–1116, 2016.

[7] Joseph N Paulson, O Colin Stine, Héctor Corrada Bravo, and Mihai Pop. Differential abundance analysis for microbial marker-gene surveys. *Nature methods*, 10(12):1200–1202, 2013.

[8] Naomi Altman and Martin Krzywinski. Points of significance: Sources of variation. *Nature methods*, 12(1):5–6, 2015.

[9] Marie-Agnès Dillies, Andrea Rau, Julie Aubert, Christelle Hennequet-Antier, Marine Jeanmougin, Nicolas Servant, Céline Keime, Guillemette Marot, David Castel, Jordi Estelle, et al. A comprehensive evaluation of normalization methods for illumina high-throughput rna sequencing data analysis. *Briefings in bioinformatics*, 14(6):671–683, 2013.

[10] Benjamin M Bolstad, Rafael A Irizarry, Magnus Åstrand, and Terence P. Speed. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19(2):185–193, 2003.

[11] Paul J McMurdie and Susan Holmes. Waste not, want not: why rarefying microbiome data is inadmissible. *PLoS Comput Biol*, 10(4):e1003531, 2014.

[12] Mark D Robinson and Alicia Oshlack. A scaling normalization method for differential expression analysis of rna-seq data. *Genome biology*, 11(3):1, 2010.

[13] Mark D Robinson, Davis J McCarthy, and Gordon K Smyth. edger: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140, 2010.

[14] Michael I Love, Wolfgang Huber, and Simon Anders. Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome biology*, 15(12):1, 2014.

[15] Viktor Jonsson, Tobias Österlund, Olle Nerman, and Erik Kristiansson. Statistical evaluation of methods for identification of differentially abundant genes in comparative metagenomics. *BMC genomics*, 17(1):1, 2016.

[16] Peter McCullagh and John A Nelder. *Generalized linear models*, volume 37. CRC press, 1989.

[17] Samprit Chatterjee and Jeffrey S Simonoff. *Handbook of regression analysis*, volume 5. John Wiley & Sons, 2013.

[18] Erik Kristiansson, Philip Hugenholtz, and Daniel Dalevi. Shotgunfunctionalizer: an r-package for functional comparison of metagenomes. *Bioinformatics*, 25(20):2737–2738, 2009.

[19] Bent Jorgensen. *The theory of dispersion models*. CRC Press, 1997.

[20] Julian J Faraway. *Extending the linear model with R: generalized linear, mixed effects and nonparametric regression models*, volume 124. CRC press, 2016.

[21] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the royal statistical society. Series B (Methodological)*, pages 289–300, 1995.

[22] James H Bullard, Elizabeth Purdom, Kasper D Hansen, and Sandrine Dudoit. Evaluation of statistical methods for normalization and differential expression in mrna-seq experiments. *BMC bioinformatics*, 11(1):1, 2010.

[23] J Zyprych-Walczak, A Szabelska, L Handschuh, K Górczak, K Klamecka, M Figlerowicz, and I Siatkowski. The impact of normalization methods on rna-seq data analysis. *BioMed research international*, 2015, 2015.

[24] Simon Anders and Wolfgang Huber. Differential expression analysis for sequence count data. *Genome biology*, 11(10):1, 2010.

[25] Junjie Qin, Yingrui Li, Zhiming Cai, Shenghui Li, Jianfeng Zhu, Fan Zhang, Suisha Liang, Wenwei Zhang, Yuanlin Guan, Dongqian Shen, et al. A metagenome-wide association study of gut microbiota in type 2 diabetes. *Nature*, 490(7418):55–60, 2012.

[26] Shinichi Sunagawa, Luis Pedro Coelho, Samuel Chaffron, Jens Roat Kultima, Karine Labadie, Guillem Salazar, Bardya Djahanschiri, Georg Zeller, Daniel R Mende, Adriana Alberti, et al. Structure and function of the global ocean microbiome. *Science*, 348(6237):1261359, 2015.