Undergraduate Thesis

Evaluating classification methods based on microbial community composition

Johannes Oskar Effenberger

Examiner: Prof. Dr. Rolf Backofen
Advisers: Paul Zierep

University of Freiburg

Faculty of Engineering

Department of Computer Science

Chair/Laboratory for Thesis Templates

March 19th, 2024

Writing Period

19. 12. 2023 – 19. 3. 2024

Examiner

Prof. Dr. Rolf Backofen

Advisers

Paul Zierep

Declaration

Abstract

foo bar

Zusammenfassung

German version is only needed for an undergraduate thesis.

Contents

List of Figures

List of Tables

List of Algorithms

1 Introduction

1.1 Introduction

Microbial community compositions in form of Operative Taxonomic Units (OTU) tabes give detailed Informations about the taxonomic composition of a given Sample. Databases like MGnify of ENA contain hundrets of thousands of Samples aswell as the associated analysis in multple forms. One of them being OTU tables. Working with such tables can quickly lead to large collections of data and finding meaningful differences in all those reads can quckly become a near impossible task for a human. Combined with the ever growing mass of samples machine learing can be a great tool to differenciate samples based on taxonomic assignments. Databases like MGnify contain samples from diferent studies and Pilepine version sorted into diferent bioms. Wile this allows easy and public access to all those samples, comparability between samples can vary based on date of creation pipeline version or sampling depth of the analysis. Preprocessing data can be used to make samples more comparable and incease perfomance of classification based on the given samples. Creating Classification models based on taxonomic assignments have many usecases from disease detection (EXAMPLE DEEP MICO) to finding outliers in datasets or varifying databases (FIND SOURCE). Many approches to creating classification Models vary in the preprocessing and classification methods used. This thesis tries to find a guideline, discribing wich methods improve the classification results based on Biom data from MGnify.

1.2 Scientific relevance

Due to its nature of complex relations and an evergrowing mass of data, Microbiology and Biology in general is a prime field for using machine learing, also visable in the many approches to use machine learing in bioinformatics. While machine learing methods oftern produce great results they also require understading or knowlage in computer science to be used efficently. This results in a barrier of entry for peaople without connections to computer scienctists. (THESIS GIVES ACCESS to CLASSIFIATTION FOR MORE PEOPLE)

2 Related Work

Give a brief overview of the work relevant for your thesis.

3 Background

Explain the math and introduce notation.

4 Resources and Methods

4.1 Resources

Resources used in this thesis are the MGnify database and the European Galaxy servers. MGnify is used as source for the taxonomic assignments in form of OTU tables and Galaxy is used to run a Workflow, which takes Biomes as input and uses DeepMicro to calculate classification models for the given biomes.

4.1.1 MGnify

MGnify is part of the European Bioinformatics Institute (EBI) and provides the service, to create an analysis of a sample. Additionally does MGnify store thousands of all ready analyzed samples and the matching analysis. Those samples and analyses are publicly accessible and thus work as a great source for around 350.000 samples with 450.000 analyses assigned to 500 different biomes. Additionally MGnify provides an access token for the European Nucleotide Archive (ENA), where more metadata about most samples is accessible. Samples on MGnify come from different studies from all over the globe and from all types of experiments. In addition to that is MGnify improving their Pipeline, resulting in Samples, differiencating themselves in origin, sampling method, pipeline version and sampling depth. [DEPENDING on PCA of Pipeline and Study] Still, especially for large biomes, there is enough diversity in samples to allow classification based on biome, instead of study pipeline version.

4.1.2 Galaxy

Usegalaxy (Galaxy) is a open-source platform, allowing the setup of servers, allowing users to run tools and workflows through a web-interface on a server. Galaxy is specialized in Bioinforatics, allowing access to many tools, written in different programming languages and allows combined usage between those. In addition to base Tools, galaxy allows the creating of workflows as well as the usage of python in form of Jupyter Notebooks to create more complex tools. With that Galaxy gives users access to tools without requiring them to be proficient in all details of the tool. Similarly users without the required hardware gain access to those tools via Galaxy. Especially in calculation heavy fields like machine learning, it eliminates the requirement for a high end GPU and CPU on the user site.

4.2 Methods

4.2.1 Gathering data from MGnify

[Here or in WF part?]

4.2.2 Preprocessing

Preprocessing in commonly used in handling data for machine learing tasks. Preprocessing describes the processing of data in with the intention of crating a better input for a following process[anders wort]. Preprocessing can include filtering of input data, reducing the count of different features in samples (feature Reduction),normalization or standardisation. The main goal of preprocessing in regards to classification of microbial community compositions is increasing classifier performance. Using samples from MGnify for biomes results in a collection of samples from different studies and pipeline versions, making them harder to compare based on the taxa assignments.

Normalization can be used to make samples comparable. Another problem in raw data is the amount of features for each sample. Because of detailed taxonomic assignments there are more features than samples for most biomes, with a lot of features only being present in a fraction of all samples. Autoencoders and only taking reads from specific taxonomic ranks can reduce the feature space of samples.

Taxa Selection

A OTU table from MGnify often all ready contains a hundred different taxonomic assignments for a single sample. Depending on the number of samples and variation in the biome, a table containing all samples of a biome can reach thousands of features. Most of these features are only contained in a few samples and such play a small role in classifying a sample to a biome. Assuming that all samples of a biome are similar, it is fair to assume that different samples of one biome follow a similar distribution on a taxonomic level. To verify this assumption the table of taxonomic assignments can be filtered to only contain occurrences of a specific taxonomic level. This results in 8 different tables (Superkingdom, Kingdom, Phylum, Class, Order, Family, Genus, Species). If all 8 Tables are joined in one Table, the table would contain a distribution of all samples for all taxonomic ranks.[Example?]

Table Normalization

Normalization of tables allows better classification. There are two main reasons to use classification. The fist is the in cease in comparability between samples. Due to the structure of MGnify, samples are collected over the years resulting in differences in collection of the probe, analyses depth and pipeline version of the analyses. Normalization samples reduces differences between number of assignments per sample and can offset small differences between samples. Another reason for normalization is, that normalization methods can change the distribution of features in a samples. For example, due to the large differences in assignments in a single sample, softmax sets most assignments to zero, while only highlighting the most pronounced features. With that, different normalization methods can express different features in a given sample.

Relative Abundance Relative abundance (also TTS) divides each count by the sum of all counts in a sample. This changes each value to its percentage proportion in the samples. Relative abundance does not change the relation between features and only affects one sample at a time. Relative abundance, together with the raw counts, build a base to compare other normalization methods against. With the difference to the raw counts being, that the total amount of assignments in a sample are replaced with the relative abundance of each sample. Making Samples with different total read counts comparable.

Softmax Softmax normalization work by [FORMUALR?] dividing the exponential value of a count by the sum of exponential values for all counts in the sample. Softmax normalization results in most values being zero, and only the highest counts in a sample remaining. One problem of softmax normalization is, that high counts can result in overflow errors in the calculation of the exponential function. To prevent overflow errors from happening, the Relative abundance table, multiplied by 100 can be used as input. This prevents overflow errors while keeping the characteristics of a softmax normalization. Softmax normalization can be used to compare the most common taxonomic assignments in different biomes.

CSS Cumulative Sum Scaling (CSS) normalization is implemented using the meagenome-Seq package from Bioconductor. In difference to softmax and relative abundance is CSS a from of library normalization. CSS works by sorting all samples by size and calculating a library wide quantile which is used to calculate a normalization factor for each individual sample. CSS is build on the assumption, that the distribution of the least common features in each sample is the same across all samples. This results in a normalization, where the normalization of each individual sample is made depending on the whole library.

RLE Relative log Expression (RLE) is also a form on library normalization, based on the assumption, that most features in a sample do not full fill a significant role to classification. For that a library wide reference sample gets calculated which is based on the mean of all samples. The median of relations between this sample to all other samples results in a scaling factor, which is used to scale all other samples.

TMM Just like CSS and RLE is Trimmed mean of M values (TMM) a form of library normalization. TMM calculates scaling factors in relation to a reference sample (under exclusion of the most expressed features). This normalization factors are used together with the samples to create a normalized matrix.

TMM and RLE are used to normalize Gene libraries. Both normalization methods are based on the assumption, that most genes do not significantly differentiate themselves between two experimental stages, meaning they are not differentially expressed(DE).

Library Normalization [IDEA] Library normalization methods (CSS,TMM,RLE) work by calculation scaling factors based on a given library of samples. While those work great for test/training sets, new samples used in classification methods are not part of the original library. Yet new samples should still be able to be normalized and classified. All three normalization methods calculate a normalization factor based on the whole library which is used to rescale all samples. These scaling factors can be applied to a new sample, normalizing it the same way as the training set. The new sample is not able to influence the scaling factors. This should not be to problematic because all scaling factors are calculated using median, average or quantiles across the whole library, resulting in a small influence of a single sample. This would only be problematic, when the new sample is very different to all ready existing samples, which would be a problem, even without library normalization.

4.2.3 DeepMicro

All used Encoders as well as Classification methods come from DeepMicro[Paper reference]. In DeepMicro they where used to predict diseases on human patients.

AutoEncoding

Autoencoders are used to encode an Input in a way, that an decoder can reconstruct the Input with as minimal loss as possible. In this process the encoded version of the input has normally a lower dimension compared to the input. In machine learning Auto encoders can be used to reduce the feature count, by using the encoded version of samples as input.

Autoencoders are a type of neural network, which can be trained on Input data to learn an effective encoding, usable for further usage. Beeing a neural network, there are many ways to influence such an encoding. DeepMicro comes with a shallow autoencoder(SAE), deep autoencoder(DAE), variational autoencoder(VAE) and convolutional autoencoder(CAE). All of them got optimized using gridsearch in the original DeepMicro paper[SOURCE].

AutoEncoding 1 SAE

A shallow auto encoder is constructed from an Input layer, latent layer and output layer. It learns a simple encoding, usable to reduce the featured of samples. Due to its small size, it is limited in encoding relations between features and mainly used for feature reduction.

AutoEncoding 2 DAE

A deep auto encoder is has, in addition to input, latent and output layers hidden layers in the encoding and decoding networks. These hidden layers allow the deep auto encoder to encode relations between features, resulting in the ability to learn a more complex encoding, compared to a shallow auto encoder.

AutoEncoding 3 VAE

Variational autoencoders (VAE) are a type of generative model, allowing generation of new samples based on a trained representation. Variational autoencoder work by learning a probabilistic representation of input samples through encoding of mean and variance, generating a destribution from which new samples can generated. The encoded mean and variance make up the latent space of the variational autoencoder, resulting in a lower dimansionality compared to the input samples.

AutoEncoding 4 CAE

Convolutional autoencoder differentiate themselves from other autoencoders by using convolutional layers instead of feed forward layers. In each layer a filter and local region of the layer before is used to calculate a part of the new layer. Convolutional layers allow regional close features to influence each other and are often used in areas where proximity of features play an important role like image or sound generation.

Convolutional layers require a Matrix as input, instead of the feature vector. For that the vector is reshaped into a $d\ x\ d\ x\ 1$ matrix with $d = \lceil \sqrt{b} \rceil$, filling empty spaces with zero. For the decoding part of the autoencoder deconvolutional (convolutional transpose layers) are used.

In DeepMicro grid search was used to optimize the number of convolutional layers as well as the number of filters in the first convolutional layer. And the number of filters got halved each subsequent layer.

Classification

Classification 1 Support vector machine

Support vector machines are a Machine learning algorithm, which Takes N dimensional Input and tries to find a separating hyperplane of dimension N-1 which separates two Classes, allowing for classification in relation of an input to the hyperplane. Support vector machines are commonly used in classification biological Data[SOURCE]. Still, there are two problems with support vector machines. The first one is that SVM perform worse when there are a lot of features, which do not have an influence on classification. This problem should be reduced by Reducing the amount of Features through focusing on a specific taxonomic rank or the usage of Autoencoding methods. Secondly SVM are not able to learn complex relations between features.

In the Original Paper Gridsearch was used to tune penalty parameter C and the RBF kernel coefficent to optimize the method.

Classification 2 Random Forest

[Bootstraping]

Similar to Support vector machines, Random Forest(RF) is often used in Microbiom analysis [Source]. Random Forest is based on Decision Trees. A decision Tree is a machine Learning algorithm where the Training data is used to Build a Decision Tree by splitting the Data on each node until there is a classification on each Leaf. Especially when using numbers as input decision trees tend to over fit. A Random Forest is using multiple Decision Trees to create a way of classifying Data. In a

Random Forest each Tree is build by only using a Random choice of Input features resulting in a lot of different decision Trees. When classifying a new sample, the sample is classified by each Decision tree and the Final classification is based on the majority of trees.

In Deep micro grid search was used to tune number of trees, minimum number of samples in a leaf node as well as the Split criteria.

Classification 3 MLP

Multilayer Perceptrons(MLP) is a feed-forward artificial neural network. A MLP is constructed from an Input layer, one or multiple hidden layers and an Output layer. At each layer Output from the layer before and weights are used in a activation function to calculate a new Output. The First layer(Input layer) takes features as input and the last layer (Output layer) outputs a classification. In case of DeepMicro only a few hidden layers where used. Increasing the number of hidden layers allows the artificial neural network to learn complex relationships between input features. In case of microbiome composition in form of Taxonomic assignments, a shallow neural network is enough to classify different biomes. Increasing the number of hidden layers is likely to result in over-fitting. Due to the limited relation between different features [Only thought maybe change or remove later]

In DeepMicro Gridseach was used to optimize Number of hidden layer, number of hidden units in the first layer, dropout rate and number of epochs.

5 Experiments

5.1 Biomes

5.1.1 Biome Selection

Finding the perfect classifier for any two biomes is hard, especially when taking into consideration the choice of biomes available on MGnify. In this thesis two groups of Biomes will be considered. Group one contains the Biomes 'Subgingival plaque' (root:Host-associated:Human:Digestive system:Oral:Subgingival plaque) and 'Supragingival plaque'(root:Host-associated:Human:Digestive system:Oral:Supragingival plaque) and Group two contains the biomes 'Soil'(root:Environmental:Terrestrial:Soil) and 'Marine' (root:Environmental:Aquatic:Marine). In the following group one will be referenced as close biome and group two as distant biomes.

Using different groups of biomes allows comparison between results. Under the assumption that close biomes have smaller differences and distant biomes have larger differences in the taxonomic assignments using both groups shows how methods perform on different datasets.

Biome Differences

[Genaue Erklärung von Plaque arten?]

[table with studys, samples for each biome?]

[Count of unique taxa in each sample biome? (Close Biomes: 3170 Features total,B1: 2435, B2:2176)]

5.2 Classifier Selection

[Needed new Result table] To allow comparison between different Classifier methods and to see how taxaselection, normalization and autoencoders influence the classifier scores, a baseline of scores is needed. To produce a baseline all three classifier are applied to the count Tables of both biome Groups. This not only gives a baseline but also allows comparison of classifier methods based on biome distance.

[Insert result Table]

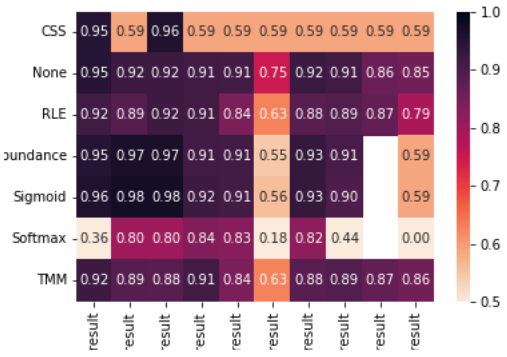
Comparing the performance of the three classifier methods against each other with Count Tables of both groups as input, F1-Scores of Random Forest outperform Support vector machines as well as the multilayer perceptron. Not only is the F1-Score higher for Random forest, but the runtime is also significantly shorter.[TODO: More info on runtime]

[Runtime comparison Table]

Due to its better performance and faster runtime, only random forest will be continued to be used in the next steps.

5.3 Taxa Depth and Normalization Selection

All normalization methods are dependent on the total count of features in each sample. This means that normalization methods perform differently depending on the size of



(a) Taxonomic Rank and Normalization in Distant Biomes

the feature space. Limiting the feature space to a specific taxonomic depth has large impact on the feature space and can, in contrast to autoencoders, be used before normalization of samples. To compare normalization methods and taxonomic depth against each other, RF is used to classify all combinations out of taxonomic ranks and normalization methods for both biome groups.

[SELECT BEST NORM / TAXA PAIRS]

Comparing classification of both biome groups, the close biomes result in better scores compared to the distant biomes.

5.4 Preclude Classification based on Study or Pipeline version

The close biome Group contains less samples on MGnify compared to the distant biome group, making it reasonable to assume that the difference in classification can be attributed to a small number of studies in the close biomes or a difference in pipeline version between both bioms. To verify if those are the reason for the high classification scores, a PCA plot can be used to test this assumptions.

PCA [definition pca explanation what it does?]

Looking at the PCA plot for Sigmoid normalization on phylum level, the combination with one of the best F1-score for the close group, shows that both biomes are not clearly separable in two dimensions. Neither studies nor pipeline version get clearly separated in the PCA plot. Additionally both bioms contain multiple studies while only Subgingival plaque contains analyses from pipeline version 5.0. Since there are multiple studies [?] in both biomes and those studies disperse differently in the PCA plot, it is fair to assume that reason for the high F1-score in RF-classification is not a low number of studies in one or both biomes.

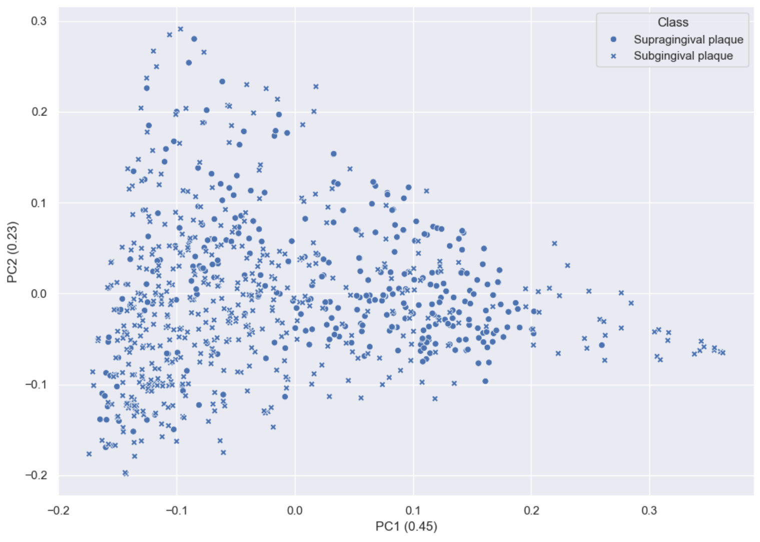
When comparing pipeline versions of all samples in the close biome group, Subgingival plaque contains samples from two pipeline versions while Supragingival plaque only contains samples from pipeline version 4.1. [Small tabe with index biom, header pipline and showing number of samples with each combination?]

Still, only about 1/3 [number] of the Subgingival plaque samples are from pipeline version 5.0. And while some separation is identifiable between both biomes. There is no clear separation between both pipeline versions. When only considering samples from Subgingival plaque there is no clear the difference between both pipeline versions, showing that a classification of both biomes is not just based on pipeline version.

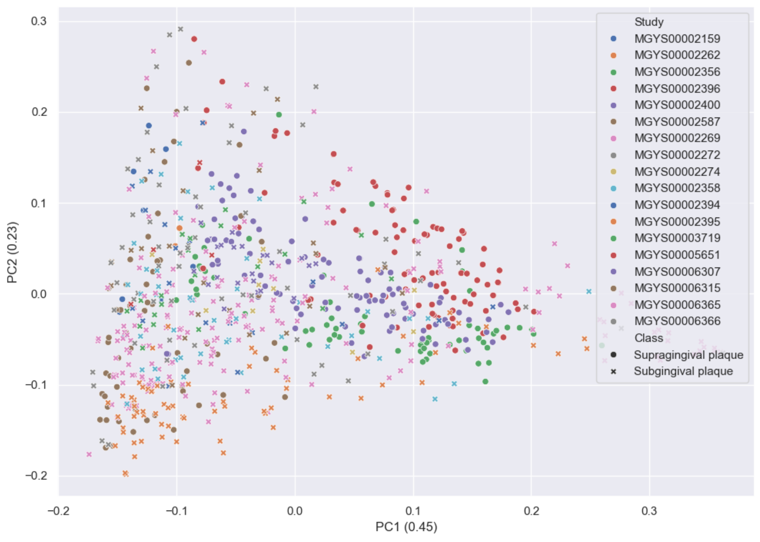
A clearer separation based on pipeline version is visible, when creating a PCA plot of the count tables normalized with relative abundance. This is also the combination with the highest F1-score for the close biomes. Any while in this graph, a clear grouping of pipeline version 5.0 is visible this grouping matches with the densest area of samples with pipeline version 4.1 of the same biome.

[Better visible separation in RA,Count than in Sig,Phy]

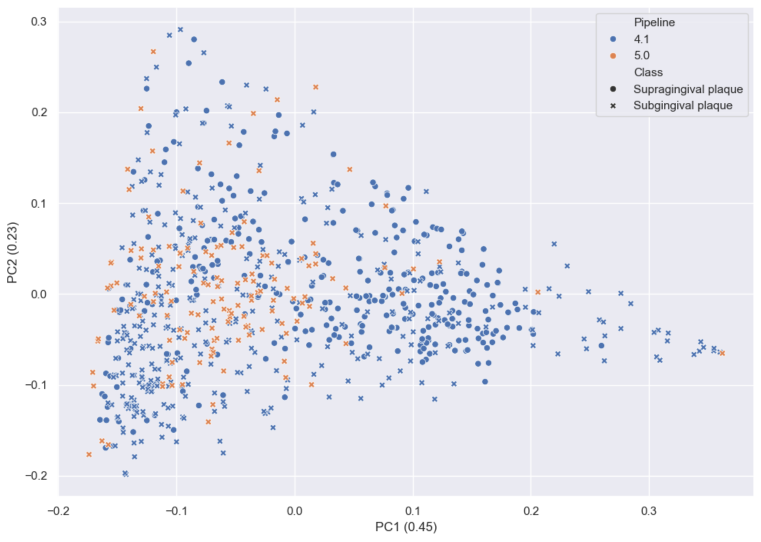
[move to end of block and add some summary part?] While it is impossible to rule out that the difference in pipeline version increases the performance of the classification, is is not the main reason for high classification scores.



(a) PCA plot for Sigmod Normalization



(b) PCA plot for Sigmod Normalization whith studies



6 Conclusion

7 Acknowledgments

First and foremost, I would like to thank...

- advisers
- examiner
- person1 for the dataset
- person2 for the great suggestion
- proofreaders

Bibliography