Norwegian University
of Life Sciences

Master's Thesis 2024 30 ECTS
Faculty of Chemistry, Biotechnology and Food Science

Predicting the status of sediment ecosystems around commercial fish farms from taxonomic and functional profiles

Caroline Opland
Chemistry and Biotechnology (M.Sc) - Bioinformatics

Abstract

The sediments around commercial fish farms are regularly monitored regarding the environmental condition. If the environmental condition is not sufficient, a quarantine period is imposed on the fish farms. It is therefore important that the environmental state is mapped quickly and with good precision. Currently, this is determined manually from inspection of sediment samples by expert taxonomists, who determine an environmental index based on macrofauna. The AQUAeD project (2021-2025) – On-site monitoring of aquaculture impact on the environment by open-source nanopore eDNA analysis – aims to replace the current environmental monitoring analyses with digital DNA based solutions, as well as moving the analyses to the facilities to achieve fast and accurate results.

This thesis is a part of the AQUAeD project, and the data used was from 16S sequencing. From the 16S data, taxonomic profiles can be made, and the essential aim of this thesis is to predict the ecosystem status from this. However, the taxonomic diversity present in sediments is significant, with numerous organisms being unidentified and lacking names within the existing taxonomy. From the metagenome data, functional profiles can be derived. This entails coding genes and categorizing them into functional groups such as EC functions, KO functions and MetaCyc pathways. Then functional profiles can be constructed accordingly.

An indicator of the ecosystem status is the nEQR values, which is what has been predicted in this thesis for both the taxonomic and functional profiles. The results from this shows that the predictions are good for both taxonomic and functional profiles, but also that the functional profiles do not give better predictions. Rather, they are very similar to each other.

In this thesis, AI (Artificial Intelligence) was used to assist with the coding, as well as finding sources for the background information in the introduction of this thesis. The AI instruments used in this thesis was ChatGPT and Perplexity.

Sammendrag

Sedimentene rundt kommersielle fiskeoppdrettsanlegg blir jevnlig overvåket med tanke på miljøtilstanden. Hvis miljøtilstanden ikke er tilstrekkelig, blir det pålagt karantenetid for oppdrettsanleggene. Det er derfor viktig at miljøtilstanden kartlegges raskt og med god presisjon. For øyeblikket blir dette bestemt gjennom inspeksjon av sedimentprøver utført av eksperter på taksonomi, som fastsetter en miljøindeks basert på makrofauna. AQUAeD-prosjektet (2021-2025) – Overvåkning av akvakulturens påvirkning på miljøet ved hjelp av open source for nanopore eDNA analyse – har som mål å erstatte de nåværende miljøovervåkningsanalysene med digitale DNA-baserte løsninger, samt å flytte analysene til fiskeoppdrettsanleggene for å oppnå raske og nøyaktige resultater.

Denne avhandlingen er en del av AQUAeD-prosjekter, og dataene som ble brukt var fra 16S sekvensering. Fra 16S dataene kan det lages taksonomiske profiler, og det essensielle målet med denne avhandlingen er å forutsi økosystemets tilstand fra dette. Den taksonomiske mangfoldigheten i sedimentene er derimot betydelig, med mange organismer som ikke er identifisert og som mangler navn innenfor den eksisterende taksonomien. Fra metagenom-dataene kan funksjonelle profiler utledes. Dette innebærer kodende gener og kategorisering av dem i funksjonelle grupper som EC funksjoner, KO funksjoner og MetaCyc pathways. Videre kan funksjonelle grupper konstrueres.

En indikator på økosystemets tilstand er nEQR-verdier, som er det som har blitt predikert i denne avhandlingen for både de taksonomiske og funksjonelle profilene. Resultatene fra dette viser at prediksjonene er gode for både taksonomiske og funksjonelle profiler, men de funksjonelle profilene gir heller ikke bedre prediksjoner. Tvert imot er de veldig like hverandre.

I denne avhandlingen ble KI (Kunstig Intelligens) brukt til å hjelpe med kodingen, samt å finne kilder til bakgrunnsinformasjonen i innledningen til denne avhandlingen. KI-instrumentene som ble brukt i denne avhandlingen var ChatGPT og Perplexity.

Acknowledgements

This thesis is a part of the AQUAeD project, and it was a part of The Norwegian University of Life Sciences (NMBU) master's program in Chemistry and Biotechnology at the Faculty of Chemistry, Biotechnology and Food Sciences (KBM) at NMBU the spring of 2024.

I would like to thank my main supervisor, Lars Snipen, for his engagement, fast responses, discussions, feedback, and the support throughout this thesis.

Lastly, I would like to thank my family for their support and motivation through five years at NMBU, especially my mother.

Table of Contents

List of Abbreviations

ASV Amplicon sequencing Variant

BIOM Biological Observation Matrix

CLR Centred Log Ratio

DADA Divisive Amplicon Denoising Algorithm

DNA Deoxyribonucleic Acid

EC Enzyme Commission

eDNA Environmental DNA

EQR Ecological Quality Ratio

HPC High-Performance Computing

ITS Internal Transcribed Spacer

IUBMB International Union of Biochemistry and Molecular Biology

KEGG Kyoto Encyclopaedia of Genes and Genomes

KO KEGG Orthologs

LASSO Least Absolute Shrinkage and Selection Operator

ML Machine Learning

nEQR normalized EQR

NGS Next Generation Sequencing

NIVA Norsk institutt for vannforskning

NMBU Norwegian University of Life Sciences

NSTI Nearest-Sequenced Taxon Index

OR Odds Ratio

OTU Operational Taxonomic Unit

PCA Principal Components Analysis

PICRUSt Phylogenetic Investigation of Communities by Reconstruction of Unobserved States

PLS Partial Least Squares

RNA Ribonucleic acid

rRNA ribosomal RNA

SRA Sequence Read Archive

TSS Total Sum Scaling

WGS Whole-Genome shotgun Sequencing

1. Introduction

The sediments around commercial fish farms are regularly monitored regarding the environmental condition. If the environmental condition is not good enough, a quarantine period is imposed on the fish farms. It is therefore important that the environmental state is mapped quickly and with good precision. Currently, this is determined manually from inspection of sediment samples by expert taxonomists, who determine an environmental index based on macrofauna. The AQUAeD project (2021-2025) – On-site monitoring of aquaculture impact on the environment by open-source nanopore eDNA analysis – aims to replace the current environmental monitoring analyses with digital DNA based solutions, as well as moving the analyses to the facilities to achieve fast and accurate results. The project will initially compare the results from traditional analyses with the results obtained by DNA-based technologies. Then, a database of DNA data will be established which can be used to describe the environmental state directly without performing the time consuming and inaccurate traditional analyses. The DNA analyses are easy to perform and therefore the entire analysis will be carried out at the facilities using cloud-based solutions for data analysis. To ensure that the technology and knowledge will benefit the entire industry, a new standard for DNA based environmental monitoring will be proposed to the authorities at the end of the project. The AQUAeD project is a collaboration between NMBU, Institute of Marine Research (Havforkningsinstituttet), Akvaplan NIVA, STIM and Aqua Kompetanse AS (Research Council of Norway, s.a.).

The study of organisms in a microbial community based on analysing the DNA within an environmental sample is called environmental metagenomics (Illumina, s.a.). Environmental metagenomics uses environmental DNA (eDNA) sequencing as a method for studying biodiversity and monitoring ecosystem changes (Illumina, s.a.). A common eDNA sequencing method is 16S ribosomal RNA (rRNA) sequencing from Illumina, which was used on the samples for this study.

1.1 Classification of environmental samples in water

The management of bodies of water is crucial for environmental purposes. The aim of the EU Water Framework Directive (EUs Vanndirektiv) is that this management follows the same principles across all of Europe. In Norway, this has been implemented in the form of “Vannforskriften”.

The main purpose of “Vannforskriften” is to provide a framework for defining environmental goals that ensure the most comprehensive protection and sustainable use of water resources. Therefore, it outlines specific guidelines for the process and criteria for the management of water resources. The environmental objective for natural surface bodies of water is to prevent any degradation in their condition and ensure that they maintain at least a good ecological and chemical status. For groundwater, the aim is to sustain at least a good chemical and quantitative status. The implementation of “Vannforskriften” requires the development of a classification system for bodies of water.

The classification system establishes specific boundaries of classes for various chemical, physical and biological parameters which are relevant for the environmental conditions in lakes, rivers, coastal waters and groundwater. This, combined with data monitoring and expert evaluations, forms the foundation for determining the overall status of ecological and chemical status of bodies of water.

The ecological status of surface water reflects the current environmental status of the body of water, including the composition of species, structure and the functioning of the ecosystem. The fundamental principle of the classification system is to categorize the ecological status of a body of water based on elements of biological quality, supported by physical and chemical conditions as supplementary parameters.

1.1.1 Classification of ecological status

The classification of ecological status of bodies of water is based on biological, physical, chemical and hydromorphological quality elements, and it contains five classes of status: “Svært god” (very good), “god” (good), “moderat” (moderate), “dårlig” (bad) and “svært dårlig” (very bad). The status “svært god” is also called the reference status and is defined as the condition of a quality element where there is little to no human impact on the body of water.

To classify the ecological status, there has been developed indices for every biological quality element that is suitable for measuring the response to a specific impact. The establishment of class boundaries involves utilizing dose-response curves that illustrate the relationship between the index response and the impact it addresses. To measure the deviations from the reference status, the Ecological Quality Ratio (EQR) is calculated, representing the ratio between observed values and water type specific reference values for the specific parameter or index. The EQR ranges from 0 to 1, with 1 indicating the best (reference status). The class status very

good/good represents the lower limit for bodies of water in the reference status, while the good/moderate class status indicates the environmental goal for the given type of water.

The ecological status of a body of water is determined by evaluating the quality element associated with the poorest class of status (or the lowest EQR value) in relative to various impacts. This follows the worst-case principle ("one out, all out") which aims to prevent the oversight of any impacts and to protect the most sensitive quality element from different influences (the precautionary principle).

The EQR value is the observed value divided by the reference value. To implement the "one out, all out" principle, it is essential to ensure comparability among EQR values for various quality elements. Therefore, the normalized EQR (nEQR) is calculated (Vannportalen, 2018).

1.2 Metagenomics

A metagenome is the collective genome of an entire microbial community. Metagenomics involves analysing the genomes found in such a community. In essence, metagenomics offers a novel approach to examining the totality of the genomic material present within a particular environment through the application of functional gene screening or sequencing analysis (Zhang et. al., 2021). There are currently two main approaches for analysing microbial communities using high-throughput sequencing: marker gene sequencing and whole-genome shotgun sequencing (WGS). The aim of WGS is sequencing all genomes existing in an environmental sample to study the biodiversity and functional capabilities in the microbial community. This allows for the characterization of the complete diversity in a habitat, including archaea, bacteria, eukaryotes, viruses and plasmids, in addition to its gene content. In comparison, marker gene analyses rely on sequencing a gene-specific region to unveil the diversity and the composition of specific taxonomic groups present in an environmental sample. The principal marker genes utilized in microbial ecology are the 16S rRNA gene (to analyse the presence of archaea and bacteria), the internal transcribed spacer (ITS) region (to analyse the composition of fungi) and the 18S rRNA (to study the presence of eukaryotes) (Pérez-Cobas, et. al., 2020).

There are advantages and disadvantages in both methods. The primary benefit of WGS, in contrast to marker gene sequencing, lies in its ability to characterize both the genetic and genomic diversity of the analysed community. Additionally, WGS allows for studying the functional capabilities present in the microbial community. Furthermore, using an adequate

sequencing depth in WGS, it is possible to assemble complete genomes from the metagenomic data, which provides valuable insights into the genomic diversity of microbial ecosystems. Recent methods in marker gene analyses have emerged to classify marker gene sequences at taxonomic levels below the genus, but the ability to differentiate between genomes with similar marker gene regions is still limited. In contrast, WGS allows for assigning taxonomy at more specific levels, such as species and strains. Another thing that differentiates the two methods are the cost and the efficiency of the analysis. In general, marker gene processing is faster, and the results are easier to analyse thus making it less expensive than WGS. This makes marker gene sequencing more advantageous for long-term studies including large numbers of samples. Both methods come with its set of advantages and disadvantages, and therefore it is crucial to choose the technique most suitable for the study (Pérez-Cobas, et. al., 2020).

1.2.1 16S rRNA sequencing

16S rRNA sequencing is a common amplicon sequencing technique and is a marker gene approach, which is used to identify and compare bacteria or fungi present within a given sample. Next-generation (NGS)-based 16S rRNA gene sequencing is a well-established method for comparing sample phylogeny and taxonomy from complex microbiomes or environments that are difficult or impossible to study (Illumina, s.a.)

The prokaryotic 16S rRNA gene is a phylogenetic marker gene (Langille, et. al., 2013), which is highly conserved between different bacterial species and is approximately 1500 bp (base pairs) long (Creative Biolabs, s.a.). The 16S rRNA gene is one of the most used genetic markers for several reasons. These reasons include:

(i) Its presence in almost all bacteria.
(ii) The function of the 16S rRNA has not changed over time, which suggests that random sequence changes are a more accurate measure of time (evolution).

(Janda and Abbott, 2007).

The 16S rRNA gene is frequently used to characterize taxonomic composition and phylogenetic diversity of environmental samples. However, the gene cannot directly identify functional categories (Langille, et. al., 2013).

1.2.2 Taxonomic profiling

Taxonomic profiling is a fundamental task in microbiome research with the aim to detect and quantify the relative abundance of microorganisms in biological samples (Ruschewyh, et. al., 2022). Taxonomic profiling gives an insight into the taxonomic composition of each analyzed sample, and it identifies the taxa present in a sample, as well as the estimation of relative abundances of organisms. The taxonomic profile will therefore contain a list of detected taxa, their estimated relative abundances and the various diversity indices.

Taxonomic profiling is a vast job since metagenomic samples contain genetic material of millions of different organisms from thousands of different species. There are two approaches for taxonomic profiling of metagenomic samples. One approach is using a genetic marker, such as the 16S rRNA gene for prokaryotes, and the other one uses whole genome sequencing.

The marker gene approach will only detect species that have the selected gene, and it cannot distinguish all species since some have almost identical 16S rRNA gene sequences, but it is much cheaper and more widely used. This approach will cluster reads based on their similarity to Operational Taxonomic Units (OTUs) (Danicic, et. al., 2018). One OTU-based method is VSEARCH.

OTU-based methods preclude the discrimination of sequence variants with less than 3 % dissimilarity. If the similarity threshold is increased, there will be a higher amount of false OTUs which are due to sequencing errors and not to biological variation. These limitations have been countered for by the development of algorithms that infer exact sequencing variants (Amplicon sequencing Variants [ASV]) by accounting for sequencing quality scores (Rolling, et. al., 2022). One ASV-based method is DADA2.

1.2.2.1 VSEARCH

VSEARCH is a versatile open-source and free of charge 64-bit tool for preparing metagenomics, genomics and population nucleotide sequence data. It was developed as an alternative to the USEARCH tool by Robert C. Edgar (2010) with the aim to be more accurate and faster than USEARCH.

VSEARCH facilitates *de novo* and reference-based chimera detection, clustering, full-length and prefix dereplication, rereplication, reverse complementation, masking, all-vs-all pairwise global alignment, exact and global alignment searching, shuffling, subsampling, and sorting.

Additionally, VSEARCH supports FASTQ file analysis, filtering, conversion and merging of paired-end reads (Rognes, 2016).

To identify similar sequences, VSEARCH uses a fast heuristic which is based on words shared by the query and target sequences. VSEARCH will then use dynamic programming to perform optimal global sequence alignment of the query against potential target genes. VSEARCH stands for vectorized search and the computation of the pairwise alignments, and is done in parallel using vectorisation and multiple threads (Rognes, 2016).

1.2.2.2 DADA2

DADA2 is an open-source R-package that models and corrects Illumina-sequenced amplicon errors. Currently, the most common way of addressing errors in Illumina-sequenced amplicon data is by quality filtering and the construction of OTUs. The Divisive Amplicon Denoising Algorithm (DADA) introduced a model-based approach for correcting amplicon errors without constructing OTUs and DADA2 extended and improved upon the original DADA algorithm (Callahan, et. al., 2016). The starting point for the DADA2 pipeline is a set of Illumina-sequenced paired-end fastq files that have been split up by sample where the barcodes/adapters have already been removed, and the end product is an ASV table (DADA2, s.a.). The DADA2 R package implements the entire amplicon workflow: filtering, dereplication, chimera identification and merging paired-end reads (Callahan, et. al., 2016). The advantages of the DADA2 pipeline compared to other methods are numerous. Primarily, the resolution is better; DADA2 infers exact amplicon variants (ASVs) from amplicon data, which resolves biological differences of even 1 or 2 nucleotides. Furthermore, the accuracy is of higher quality; DADA2 reports fewer false positive sequence variants than other methods report false OTUs (DADA2, s.a.).

1.2.3 Functional profiling

Functional profiling of metagenomic sequencing is a tool that provides insight into the genes that are present and not present in the data (Franzosa, et. al., 2018). Functional profiling can be done by obtaining functional categories and then assign genes to them. In this thesis functional categories are referred to as Enzyme Commission numbers (EC functions), KEGG (**K**yoto **E**ncyclopaedia of **G**enes and **G**enomes) orthologs (KO functions) and MetaCyc pathways.

The EC number is a numerical system that categorizes enzymes based on the chemical reactions they facilitate. In other words, EC numbers do not identify enzymes themselves, but rather the reactions they catalyze. The EC system include six primary levels:

1. EC 1 for Oxidoreductase reactions
2. EC 2 for Transferase reactions
3. EC 3 for Hydrolase reactions
4. EC 4 for Lyase reactions
5. EC 5 for Isomerase reactions
6. EC 6 for Ligase reactions

The EC number classification system serves as a bridge between genomics and chemistry. The EC number classification system established by the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology (IUBMB) is used by researchers to clearly reference enzymes and their functions. Enzymatic reactions or chemical information is represented by the EC numbers, but they also serve as identifiers of enzymes and enzyme genes, which offers genomic information as well. The EC number is pivotal in both categorizing enzymatic reactions and connecting enzyme genes or proteins to reactions within metabolic pathways. Consequently, this dual functionality of EC numbers enables the linking of genomic repertoires of enzyme genes to the chemical repertoire of metabolic pathways (Hu, et. al., 2012).

The KO (KEGG Orthology) database (KO (KEGG ORTHOLOGY) Database, 2023) is a database where functions on the molecular level are stored, in which each KO is defined as a functional ortholog of genes and proteins. Networks of molecular interactions, reactions and relations, such as KEGG pathway maps, BRITE hierarchies and KEGG modules, represent higher-level functions (Kanehisa, et. al., 2017). Every node within the network, like a box in the KEGG pathway map, is assigned a KO identifier (referred to as a K number), which serves as a functional ortholog derived from genes and proteins experimentally characterized in specific organisms. These identifiers are then used to assign orthologous genes in other organisms based on sequence similarity. The level of detail in defining “function” varies based on context, which results in KO groupings that can represent either highly similar sequences within a restricted organism group or a more divergent group (KO, (KEGG ORTHOLOGY) Database, 2023).

The KO database – which is a large, manually curated collection of protein families – is one of the core databases of KEGG, and it serves as fundamental reference for linking genes with

pathways through K number identifiers. Within the KO database, genes sharing similar functions are organized into ortholog groups, referred to as KO entries. Each KO contains several segments of gene information and contributes to one or more paths. Typically, KOs denote groups of genes sharing similar functions and are defined within the framework of the KEGG pathway and other molecular networks. Consequently, by assigning genes K numbers, the entire KEGG pathway and molecular networks can be automatically reconstructed. Presently, 48 % of all protein sequences are assigned to KOs within the KEGG database (Zhang, et. al., 2023).

MetaCyc Metabolic Pathway database (MetaCyc, s.a.) is an extensive database containing metabolic pathways and enzymes spanning all domains of life. The information within the MetaCyc database is grounded in evidence and meticulously curated, which makes it a comprehensive reference resource for metabolism (Caspi, et. al., 2020). Pathways related to both primary and secondary metabolism are stored in the MetaCyc database, along with associated metabolites, reactions, enzymes and genes. As of May 2024, MetaCyc contains 3153 pathways, 19 020 reactions and 19 372 metabolites (MetaCyc, s.a.).

MetaCyc serves as a reference pathway database used for predicting the pathway repertoire of an organism based on its annotated genome. MetaCyc offers a searchable encyclopaedia of enzymes and pathways, which details the catalytic functions of enzymes, among other things. The aim of the MetaCyc database is to offer a large selection of pathways sourced from various organism. The guiding principle of MetaCyc is to encode pathways documented in experimental literature. Each pathway is tagged to the organism(s) in which it has been experimentally observed, as determined by evaluations of literature up to date. However, since experimental evidence have confirmed the existence of most pathways in only a limited number of organisms in which they actually occur, and because MetaCyc does not cover all available literature, the species information within MetaCyc is incomplete. Nonetheless, it reflects wet-lab findings rather than computational determinations (Karp, et. al., 2002).

Marker-based gene sequencing, such as 16S rRNA sequencing, does not provide any information about the functional capabilities of sampled communities. In order to overcome this obstacle, PICRUSt (**P**hylogenetic **I**nvestigation of **C**ommunities by **R**econstruction of **U**nobserved **S**tates) was developed in 2013 and later improved upon with PICRUSt2 (Douglas, et. al., 2020).

1.2.3.1 PICRUSt2

PICRUSt2 is a software that is used for predicting functional abundances based on marker gene sequences (Douglas, 2021). PICRUSt (now known as "PICRUSt1") was developed for predicting functions from 16S marker sequences and it is still widely used, but it has some limitations. The required input sequences for PICRUSt1 are OTUs and because of this restriction PICRUSt1 is not compatible with ASVs, which have finer resolution. ASVs will therefore produce a more precise differentiation of closely related organisms (Douglas, et. al., 2020). PICRUSt2 also includes these improvements over the original version (Douglas, 2021):

- Allow users to predict functions for any 16S sequences. Representative sequences from OTUs or amplicon sequence variants (e.g. DADA2 and deblur output) can be used as input by taking a sequence placement approach.
- Database of reference genomes used for prediction has been expanded by >10X.
- Addition of hidden-state prediction algorithms from the `castor` R package.
- Allows output of MetaCyc ontology predictions that will be comparable with common shotgun metagenomics outputs.
- Inference of pathway abundances now relies on MinPath, which makes these predictions more stringent.

The first step of PICRUSt2 is aligning OTUs or ASVs to reference sequences (HMMER). Then the second step is placing the OTUs or ASVs into a reference tree (EPA-NG and GAPPA). The third step is inferring gene family copy numbers of OTUs or ASVs (castor). The fourth step is determining gene family abundances per sample. The fifth and last step is inferring pathway abundances (MinPath) (Douglas, 2021).

1.3 Machine Learning

Machine learning (ML) are methods that can find relationships and patterns in data, and they use historical data as input to make predictions (Tucci, 2023). One type of machine learning is supervised machine learning.

Supervised machine learning uses a known dataset that includes desired inputs and outputs. The algorithm must then find a method to determine how to arrive at those inputs and outputs. Since the operator knows the correct answers to the issue, the algorithm will identify patterns in the data, learn from observations and make predictions. One type of supervised machine learning

is regression. Using regression methods, the machine learning program must estimate and understand the relationships between variables. Regression methods are focused on one dependent variable and a series of other changing variables, which makes it particularly useful for prediction (Wakefield, s.a.). Two examples of regression methods are Partial Least Squares (PLS) regression and Least Absolute Shrinkage and Selection Operator (LASSO) regression.

1.3.1 PLS Regression

PLS regression is a multivariate statistical analysis that allows for comparison between multiple response variables, as well as multiple explanatory variables. The method was designed to deal with problems such as a small sample set in the data, missing values and multicollinearity. The aim of PLS regression is to predict one or more responses (columns in Y) from potentially many predictors (columns in X), as well as describing the ordinary structure underlying the two variables. The analysis is similar to principal components analysis (PCA) regression and multiple linear regression (Pirouz, 2006).

Predicting Y from X when Y is a vector and X is full rank could be accomplished by multiple regression, but when the number of predictors is large compared to the number of observations, X must be singular. Therefore, the multiple regression method is no longer feasible. To cope with this issue, there has been developed several approaches. One of these approaches is to perform PCA on the X matrix and then use the principal components of X as regressors on Y. The principal components are orthogonal which eliminates the multicollinearity issue, but the issue regarding the selection of an optimum subset of predictors is still remaining. A possible way of solving this issue is to keep just a few of the first components, but they are chosen to explain X rather than Y, which means there is no assurance that the principal component that explain X also hold relevance for Y. In contrast, PLS regression identifies components from X that maintain relevance for Y. PLS regression specifically searches for a set of components, known as latent vectors, which enables a simultaneous decomposition of X and Y. These latent vectors are constrained to maximize the explanation of covariance between X and Y. This process extends beyond PCA. Subsequently, a regression step utilizes the decomposition of X to make predictions of Y (Abdi, s.a.).

1.3.2 LASSO Regression

LASSO regression is a well-known method used in statistical modelling and machine learning to estimate the relationship between variables and make predictions. The primary aim of

LASSO regression is to strike a balance between model simplicity and accuracy. It accomplishes this by incorporating a penalty term into the conventional linear regression model, promoting sparse solutions where certain coefficients are compelled to be precisely zero. This characteristic makes LASSO especially advantageous for feature selection, as it can autonomously recognize and eliminate irrelevant redundant variables. (Kumar, 2023).

LASSO regression is a shrinkage method. The LASSO estimate is defined by:

$$\hat{\beta}^{\text{lasso}} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2$$

$$\text{subject to } \sum_{j=1}^{p} |\beta_j| \leq t.$$

In LASSO regression the constant β_0 is re-parametrized by standardizing the predictors; the solution for $\hat{\beta}$ *is* $\bar{y}$, and thereafter a model is fit without an intercept (Hastie, T., et.al., 2008).

The lasso coefficients, $\hat{\beta}_\lambda^L$, minimize the quantity:

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| = \text{RSS} + \lambda \sum_{j=1}^{p} |\beta_j|.$$

In statistical parlance, LASSO regression uses an ℓ_1 penalty, and in LASSO regression, the coefficient estimates are shrunk towards zero. However, with the LASSO method, the ℓ_1 penalty leads to some coefficient estimates being forced to exactly zero when the tuning parameter λ reaches a certain threshold. LASSO regression also performs a *variable selection*. The output of LASSO regression can be described as *sparse* models, meaning they entail only a subset of variables. It is crucial to select a good value of λ, and depending on this value, LASSO regression can produce a model involving any number of variables (James. G. et. al., 2021).

1.4 Aim of the Study

The aim of this study is to predict the nEQR values based on taxonomic profiling. The taxonomic predictors were produced using two different methods, VSEARCH and DADA2. The nEQR values were predicted using two different machine learning methods, PLS regression and LASSO regression. In addition to taxonomic profiling, the nEQR values were also predicted using functional profiling using PICRUSt2, and the aim of this is to investigate

whether functional profiling can provide anything beyond taxonomic profiling. In other words, the general aim of this study is to:

- First, predict the nEQR values based on taxonomic profiling from VSEARCH using the machine learning methods, PLS regression and LASSO regression.
- Then, using the results from VSEARCH to predict the nEQR values based on functional profiling from PICRUSt2 using the same machine learning methods.
- Lastly, investigate whether functional profiling can provide anything beyond taxonomic profiling.

2. Methods

The data analysis and wrangling were done using RStudio 4.3.1 (R Development Core Team, 2010) on Orion, the high-performance computing (HPC) from NMBU. The R package tidyverse (Wickham et. al., 2019) were used throughout the entire thesis, as well as the ggplot2 (Wickham, et. al., 2016) package which were used to generate the figures.

2.1 The Data

The metadata table from the AQUAeD project was loaded into R as a data frame with one row per sample and 90 columns describing the samples. There are 1630 rows in the table and thus 1630 samples. Some of these samples had missing values (NA) in the nEQR column, which contains the values to be predicted, and were therefore filtered out, which resulted in 1414 rows or samples remaining in the metadata table. All the columns describing the samples were not needed, and therefore the data frame was filtered to include the 6 columns shown in *Table 1*.

Table 1. *This table shows the most important columns from the metadata table from the AQUAeD project, as well as the description of these columns.*

Column name	**Column information**
SampleID	The ID for each sample
filename	The filename for R1 reads for each sample
filename2	The filename for R2 reads for each sample
Location	The name of the location for each fish farm where samples were taken
Station	The station on each fish farm where samples were taken
nEQR	The nEQR value for each sample

In addition, the R1 and R2 files for each sample were also provided for this thesis. These files were de-multiplexed, and the primers were removed so that they could be uploaded to the Sequence Read Archive (SRA, National Library of Medicine, s.a.).

2.2 Taxonomic Predictors

Two sets of taxonomic predictors were generated based on classic grouping of sequences, one with low and one with high resolution. The one with low resolution was produced using VSEARCH with a 95 % threshold identity, and the one with high resolution was made using

DADA2. Both set of predictors were then used in a machine learning method to predict the nEQR-index values. Since nEQR is a numeric response and there are many predictors, PLS regression and LASSO regression were utilized.

The VSEARCH and DADA2 methods were compared to each other, which shows the frequency of the library sizes. Library size refers to the total number of mapped reads (PennState Eberly College of Science, 2018).

2.2.1 VSEARCH

The first processing tool that were used on the data was VSEARCH. The processing of the raw data involves several steps, and it was separated into two separate shell scripts. The first script changes the raw fastq files of R1 and R2 reads to a set of fasta files. The steps in the first part of VSEARCH includes reading the metadata table. Then for each sample, the read-pairs are merged. The reads are then filtered based on the quality scores and lastly the reads are de-replicated. Since the exact same steps are performed on each sample, the code is repeated by using a for-loop.

Before VSEARCH is used on the data, the columns needed from the metadata table is read into the shell script using the programming language *awk*. The columns needed from the metadata table are "SampleID", "filename" and "filename2". What these columns represents can be seen in *Table 1*. In the first step where VSEARCH is used inside the for-loop, paired-end reads are aligned and merged. The two reads in a pair come from each end of the genomic fragment that were sequenced. These fragments are amplicons that were copied out of the 16S gene. Designing the primers to match at various locations yields amplicons of certain lengths. These amplicons are typically designed to be less than two times the read length, meaning that when they are sequenced from each end, there will be a region in the middle where the two reads overlap. Because of this, the two reads can be aligned and merged into one longer read spanning the entire amplicon. The output from this step is a fastq file, meaning that the quality scores for the merged sequences are still existing.

In the second step, the reads are filtered based on the quality scores. The threshold for the quality scores is set using the option `--fastq_maxee_rate`, which is set to an error probability of 0.01, corresponding to a quality score of 20.

The final step of part one in VSEARCH is de-replicating the reads. Dereplication is the process in which unique sequences are identified so that only one copy of each sequence is reported

(Bioinformatic Methods for Biodiversity Metabarcoding, s.a.). The output from this is a set of fasta files.

Part two of VSEARCH was done in a separate shell script from part one. In this part, all the reads from all the samples are first written into one large fasta file. Reads from all samples should be considered in subsequent clustering. Organisms may appear in multiple samples, and utilizing all reads yields higher read counts for each OTU, which facilitates easier clustering.

The next step is de-replication of all reads. In this part, the minimum copy number is set to 2. Reads that are only reported once in a set of reads, most likely contain sequencing errors, and these reads can therefore never form a separate cluster.

The third step of part two in VSEARCH, is the clustering of all de-replicated sequences. In this part, it is determined which reads belong together and form the abundance of each cluster. These clusters are the OTUs. The identity determines the size of all clusters and in this thesis, it was set to 95 %. The centroid sequence of a cluster is the sequence with the highest copy number within the cluster, serving as its representative sequence. Subsequently, all other members must exhibit a similarity higher than a threshold of 95 % to this centroid sequence. The output from the clustering consists of a fasta file containing these centroid sequences.

The next step is chimera filtering. The term “chimera” refers to an artifact of the PCR amplification process that was conducted before sequencing. Throughout this process, amplicons may emerge as a mixture of two original amplicons, where the first part is from one organism and the second part is from another. Reads like this should be discarded. The procedure involves exporting the non-chimera sequences to a temporary file and subsequently replacing the contents of the centroids fasta file with the non-chimera data. Consequently, the updated file should be devoid of chimeras.

The final step is assigning all reads to the OTUs. In this part, each read is compared to the centroids, and if a read exhibits similarity greater than 95 % to any centroid, it is classified as a member of that OTU.

The output from the VSEARCH processing is a table of read counts and a FASTA file with sequences. In this table, each row is an OTU, and each column is a sample. The numbers represent read counts, indicating the quantity of reads OTU has in each sample. The output is saved in a tab-delimited text file. The number of OTUs in the readcount table was 43 767.

This was done according to the BIN310 module 10 – Metabarcoding data (Snipen, L., 2023).

2.2.2 DADA2

An alternative to VSEARCH processing is the DADA2 pipeline. The DADA2 processing contains one single R script with code for processing reads with the dada2 R package, and a shell script to run this R script on Orion.

The first part of the dada2 R script contains reading the metadata table, and the filtering and trimming of reads. Instead of looping over samples, which were done with VSEARCH, the DADA2 will handle all samples in a single operation. When filtering with DADA2, the *expected error* is specified. This is the error probability multiplied by the read length. This means that instead of averaging the error probabilities, they are directly summed up. The expected error cannot exceed a certain threshold value, which in this case were set to 2.5 for the R1 reads and 5.0 for the R2 reads.

The second part of the dada2 R script contains the denoising. The main concept of dada2 is to estimate the level of sequencing error and utilize this information for sequence grouping. Only sequences that fall within the range of sequencing error differences should be grouped together. The first part of the denoising step is to accurately estimate the relationship between quality scores and variations in the sequences. This involves estimating the rate of substitutions from one nucleotide to another. The error rate model is trained separately for the R1 and R2 reads, resulting in two different error objects. The next step of the denoising, is to de-replicate, which is done separately for the R1 and R2 reads as well. Finally, the denoising is performed separately on the R1 and R2 reads. The central denoising algorithm of the DADA2 R package relies on a model that characterizes the errors present in Illumina-sequenced amplicon reads. This error model quantifies the rate $\boldsymbol{\lambda_{ji}}$ at which an amplicon read with sequence **i** is produced form sample sequence **j** as a function of sequence composition and quality. A Poisson model, characterized by the rate parameter $\boldsymbol{\lambda_{ji}}$, is applied to estimate the number of repeated observations of sequence **i**. This model is utilized to compute the p-value, indicating whether the abundance of amplicon reads for sequence **i** aligns with the null hypothesis based on the error model. The p-values serve as the dividing criteria within an iterative partitioning algorithm. This algorithm persist in dividing sequencing reads until all partitions are deemed consistent with the originating from their central sequence (Callahan, et. al., 2016).

The third part of the dada2 R script contains the merging of the R1 and R2 reads. This is done using the mergePairs()-function, and the objects used in the function are both the de-replicating

objects and denoising objects for the R1 reads, as well as the R2 reads. The output from this is an object of merged reads.

The final part of the dada2 R script consist of making the readcount table from the merged object using the makeSequenceTable()-function, and the function removeBimeraDenovo() was used for filtering chimera. The output from DADA2 was a readcount table with the same structure as VSEARCH, meaning each row is an ASV and each column is a sample. The numbers represent read counts, indicating the quantity of reads ASV has in each sample. The number of ASVs in the readcount table was 68 490.

This was done according to the BIN310 module 10 – Metabarcoding data (Snipen, L., 2023).

2.3 Functional Predictors

Based on the results from the taxonomic predictors, the PICRUSt2 tool were utilized to make alternative functional predictors. Then the same machine learning methods (PLS regression and LASSO regression) were used to get predictions of nEQR so that the predictions can be compared directly to each other.

2.3.1 PICRUSt2

PICRUSt2 on the output from VSEARCH were done using this code:

```
picrust2_pipeline.py -s seqs.fna -i \

readcount_vsearch.biom -o picrust2_out_pipeline -p 1
```

(Douglas, 2021).

The inputs from this command is the file *seqs.fna* and *readcount_vsearch.biom*. The former file contains the OTU's with their corresponding 16S rRNA gene sequences. The id for each OTU is in each of the header lines starting with ">". The latter file is a BIOM file which is binary encoded. The first column of this file contains the OTU ids, and the additional columns represent the different samples with the counts representing the number of reads within each of those samples.

The command above is easiest way to run PICRUSt2, and it automatically run all the steps that are described below.

The first step of PICRUSt2 is to insert the OTU's into a reference tree. The default of this reference tree is based on 20 000 16S sequences from genomes in the Integrated Microbial Genomes database. This step will specifically:

- Align the OTU's with a multiple-sequence alignment of reference 16S sequences with HMMER (HMMER, s.a.).
- Finds the most likely placements of the OTU's in the reference tree with EPA_NG (Barbera, et. al., 2018) or SEPP (Mirarab, et. al., 2011).
- Outputs a tree file with the most likely placements for each OTU as the new tips with GAPPA (Czech, et. al., 2020)

The command for this step is:

```
place_seqs.py -s ../seqs.fna -o out.tre -p 1 \
--intermediate intermediate/place_seqs
```

The required input is the FASTA of the OTU sequences (the *seqs.fna* file) and the key output file is the out.tre, which is a tree in the newick format of the OTU's and reference 16S sequences.

There are several approaches for inferring what the likely trait values are for unknown lineages on a phylogenetic tree. PICRUSt2 uses the approaches implemented in the castor R package (Louca and Doebeli, 2017). This step will run maximum parsimony by default. In this step, the missing genome for each OTU will be predicted, in other words it will predict the copy number of gene families for each OTU. Predictions for several gene family databases are possible, and in this thesis the predictions for EC functions, KEGG orthologs and MetaCyc Pathways were used. The number of 16S rRNA gene sequences per OTU were also predicted. The second step of the PICRUSt2 pipline is shown in the script below for EC functions:

```
hsp.py -i 16S -t out.tre -o marker_predicted_and_nsti.tsv.gz \
-p 1 -n
hsp.py -i EC -t out.tre -o EC_predicted.tsv.gz -p 1
```

The output files from these commands are marker_predicted_and_nsti.tsv.gz and EC_predicted.tsv.gz. The first output is the nearest-sequenced taxon index (NSTI) value for each OTU, which corresponds to the branch length in the tree from the placed OTU to the

nearest reference 16S sequence. The second output is the predicted copy number of all EC number for each OTU.

In the next step of the PICRUSt2 pipeline, the predicted gene families weighted by the relative abundance of OTU's in their community are produced. This output can be produced by plugging in the BIOM file of OTU abundances per samples and there are two steps performed at this stage:

- The read depth per OTU is divided by the predicted 16S copy numbers. This is done to mitigate the impact of variations in 16S copy numbers among different organisms, which could lead to interpretation challenges.
- The OTU read depths per sample (after normalizing by 16S copy number) are multiplied by predicted gene family copy numbers per OTU.

The script below will run these steps:

```
metagenome_pipeline.py -i ../table.biom -m
marker_predicted_and_nsti.tsv.gz -f EC_predicted.tsv.gz \
                        -o EC_metagenome_out --strat_out
```

The desired output file within the EC_metagenome_out from this command is EC_metagenome_out/pred_metagenome_unstrat.tsv.gz, which is the overall EC number abundances per sample.

The steps described above is to generate the EC functions, but the same can be done to get the KO functions.

The last major step of the PICRUSt2 pipeline is to infer the pathway-level abundances with pathway_pipeline.py. The default of this script is to infer MetaCyc pathway abundances based on EC number abundances. The number of steps that this script perform by default are the following:

- Regroups EC numbers to MetaCyc reactions.
- Infers which MetaCyc pathways are present based on these reactions with MinPath (Ye and Doak, 2009).
- Calculates and returns the abundance of pathways identified as present.

These steps are run with this command:

```
pathway_pipeline.py -i
EC_metagenome_out/pred_metagenome_contrib.tsv.gz \
                      -o pathways_out -p 1
```

The output of this script is in the pathways_out folder, and the desired output file are the unstratified MetaCyc pathway abundances, which are in the file named path_abun_unstrat_per_seq.tsv.gz.

The outputs from the PICRUSt2 pipeline that were used for the machine learning methods were readcount tables with EC functions (EC_metagenome_out/pred_metagenome_unstrat.tsv.gz), KO functions (KO_metagenome_out/pred_metagenome_unstrat.tsv.gz) and MetaCyc pathways (pathway_out/ path_abun_unstrat_per_seq.tsv.gz). The number EC functions in the readcount table was 2329. The number of KO functions in the readcount table was 7634. The number of MetaCyc pathways in the readcount table was 430. (Douglas, 2023).

2.4 Machine Learning

The machine learning methods that were performed on the taxonomic and functional predictors were PLS regression and LASSO regression, and the response for all of these regression methods were the nEQR values. In both machine learning methods, cross validation was implemented. Cross validation is a fundamental technique in machine learning, and it is used to evaluate a model's performance on unseen data. In cross validation, the data is split into multiple segments, where one segment is used as a validation set, while the model is trained on the remaining segments. This procedure is repeated numerous times, where a different segment is used as the validation set each time. Ultimately, the outcomes from each validation step are averaged to generate a more reliable assessment of the model's performance. Cross validation is a crucial component in machine learning methods, serving to validate the chosen model for deployment is robust and capable of effectively generalizing to new data.

The primary aim of cross validation is preventing overfitting, which is a modelling error where a model is trained too well on the training data and thus performs poorly on new, unseen data. Through assessment across multiple validation sets, cross validation offers a more accurate evaluation of the model's generalization performance, indicating its capability to perform effectively on new, unseen data (GeeksforGeeks, 2023).

In this thesis, the data set was divided into 236 different segments, where each segment contained the samples from the same station at the same fish farm, i.e. all the samples from the identical Location-Station (see *Table 1* for explanation) were in the same segment.

When using machine learning methods, there will always be prediction errors. Errors of prediction are defined as "*the differences between the observed values of the dependent variable and the predicted values for that variable obtained using a given regression equation and the observed values of the independent variable"* (Allen, 1997).

2.4.1 PLS Regression

The PLS regression on the readcount table from VSEARCH and DADA2 were done using the *plsr()* function from the R package pls (Liland, K., et. al., 2023). The response was the nEQR values from the metadata table and the predictors were the OTUs (VSEARCH) or ASVs (DADA2), which were restored in a matrix. The PLS regression was done using a cross validation in such a way that all samples from the same station from the same fish farms were in the same cross validation segment. This was stored as a list.

The numbers of OTUs/ASVs in the matrices were reduced for both VSEARCH and DADA2 to improve the PLS regression. The number of OTUs/ASVs with a prevalence of less than 1 % of the total number of samples were removed from the matrices for both VSEARCH and DADA2. This resulted in 28 548 numbers of OTUs for VSEARCH and 6875 numbers of ASVs for DADA2.

The OTUs/ASVs were normalized using Total Sum Scaling (TSS) and Centred Log Ratio (CLR). TSS refers to using the total read counts for each sample as the size factors to estimate the library size or scale the matrix counts. The count data will be TSS normalized by dividing the OUT/ASV read counts by the total number of reads in each sample to convert the counts of proportion. The total number of OUT/ASV in the sample is utilized to adjust the abundance of each OUT/ASV (Xia, 2023). The formula for TSS is:

$$Y_{*j} = \frac{X_{*j}}{\sum_{i=1}^{n} X_{i,j}}$$

(Vinje and Snipen, 2023).

CLR transformation was first introduced by Aitchison (1986) (Gloor, 2017) and it is defined as the logarithm of the components after dividing by the geometric mean of x:

$$Clr(x) = \left[In(\frac{x_1}{g_m(x)}), \ldots, In(\frac{x_i}{g_m(x)}), \ldots, In(\frac{x_D}{g_m(x)})\right]$$

Where $x = (x1, \ldots, xi, \ldots, xD)$ represent the composition, and $g_m(x) = D\sqrt{x_1 \cdot x_2 \ldots x_D \cdot}$ is to ensure that the elements of clr(x) is zero (Xia, Y.). The PLS regressions using no normalization, TSS normalization and CLR normalization were then compared to each other for both VSEARCH and DADA2 to find the most optimal PLS regression.

The predictions from the PLS regression were extracted by fitting a PLS model with the nEQR values as the response and the OTU's/ASV's as predictors. The command for doing this was:

```
plsr(y ~ X, ncomp = 20, validation = "CV", segments = seg.lst)
```

where y is a table with the nEQR values and X is a matrix with the OTU's/ASV's. The ncomp is the number of components, validation is the type of validation used in the PLS regression, which in this case was cross validation. Lastly, segments is a vector that dictate how the cross validation should be performed. In this case, the cross validation is performed in such a way that all the samples in the same Location-Station is in the same segment. This resulted in 236 segments.

Comparing the different PLS regressions were done by computing Manhattan distances, i.e. the distances between the observed and the predicted nEQR values, for each PLS regression for both VSEARCH and DADA2. The number of components utilized in the PLS regressions for both VSEARCH and DADA2 were 20 components. By computing the Manhattan distances, the lowest prediction error from the cross validation can be determined, as well as the number of components that yield the lowest value. The results from this will show which PLS regression method is the most optimal. The Manhattan distances were compared to each other by plotting a diagram with points and lines for both VSEARCH and DADA2.

The most optimal PLS regression method with the optimal number of components for both VSEARCH and DADA2 were examined by making a scatter plot that shows the observed nEQR values compared to the predicted nEQR values. The predictions from the PLS regression were found inside the PLS regression objects.

The PLS regression for the outputs from PICRUSt2 were done in a similar way as it were done for VSEARCH and DADA2 outputs. The PLS regressions were done for the readcount tables for EC functions, KO functions and MetaCyc pathways from the PICRUSt2 outputs. The PLS regressions were done without normalization, with TSS normalization and with CLR

normalization. Manhattan distances were calculated in order to find out which normalization were the most optimal the same way as for VSEARCH and DADA2.

2.4.2 LASSO Regression

The LASSO regression was done by using the function *cv.glmnet()* from the R package glmnet (Friedman, J., et. al., 2023). The predictors were the OTUs (VSEARCH) or the ASVs (DADA2) which were restored in a matrix and the response were the nEQR values from the metadata table. The LASSO regression was done using a cross validation in such a way that all samples from the same station from the same fish farms were in the same cross validation segment. This was stored as a vector.

The numbers of OTUs/ASVs in the matrices were reduced for both VSEARCH and DADA2 to improve the LASSO regression. The number of OTUs/ASVs with a prevalence of less than 1 % of the total number of samples were removed from the matrices for both VSEARCH and DADA2. This resulted in 28 548 numbers of OTUs for VSEARCH and 6875 numbers of ASVs for DADA2.

The OTUs/ASVs were normalized using CLR normalization. The predictions from LASSO regression were extracted by fitting a LASSO model with the nEQR values as the response and the OTU's/ASV's as the predictors. The command for doing this was:

```
cv.glmnet(X, y, foldid = seg.vct)
```

where `X` is a matrix with the OTU's/ASV's, and `y` is a table with the nEQR values. The `foldid` is a vector that dictate how the cross validation should be performed. In this case, the cross validation is performed in such a way that all the samples in the same Location-Station is in the same segment, the same as for PLS regression. This resulted in 236 segments.

The Manhattan distances, i.e. the differences between the observed and the predicted nEQR values, were computed for the LASSO regressions as it was for the PLS regressions.

The predictions from the LASSO regression were derived by using the function *predict()* from the glmnet R package. It is not clear whether this function will give the actual predictions of interest. In order for it to be a prediction:

- The model must be trained on a training set.
- The trained model must be used to make predictions, but the data has to be from a test set.

Therefore, the cross validation was done manually by making a for loop that both trains and predicts the nEQR values. First, the results from the *cv.glmnet* object were used to make a new matrix of selected predictors. This picks out the columns from the X matrix (OTU's/ASV's) that was selected and puts them in a table with y (the nEQR values), as well as the segments. The *lm()* function is then used to customize a model. Then, by using the same data as test data, artificial "predictions" can be made. For this the training data will be reused as test data. Now, the cross validation can be run quite explicitly by using a for loop that trains and predicts the nEQR values.

The predictions were then evaluated by making a scatter plot that shows the observed nEQR values compared to the predicted nEQR values.

The LASSO regression for the outputs from PICRUSt2 were done in a similar way as to the LASSO regression that were done on the outputs from VSEARCH and DADA2. The LASSO regression was done using CLR normalization on the readcount tables with EC functions, KO functions and MetaCyc pathways.

After the LASSO regressions, the PLS regressions were performed again on the OTU's/ASV's/Functions chosen from the LASSO regressions.

2.4.3 Fisher Exact Test

Fisher Exact Test is a statistical test, which is used to determine whether the proportions of categories in two group variables significantly differ from each other (StatsTest, 2024). The p-value in Fisher Exact Test indicates the probability of observing the data, i.e. the genera that causes pollution, if the null hypothesis is true. These are the hypothesises:

- **H_0:** There is no association between the genus and the cause of pollution. In other words, the proportion of cases where the genus is associated with pollution is the same as the proportion where it is not associated with pollution. Mathematically, it could be stated as this: the probability of a genus causing pollution is equal to the probability of a genus not causing pollution.
- **H_1:** There is a significant association between the genus and the cause of pollution. This means that the proportion of cases where the genus is associated with pollution is significantly different from the proportion where it is not associated with pollution.

If the p-value is below 0.05, it suggests that the association between the variables is statistically significant. In other words, there is enough evidence to reject the null hypothesis in favour of the alternative hypothesis.

3. Results

3.1 The Data

The dataset from the AQUAeD project contains 1414 samples with valid nEQR values (nEQR values that are not "NA"). The nEQR value is a continuous and numeric value, and it is between 0 and 1 for each sample, which gives these 5 nEQR categories:

- 1.0-0.8 – "Svært god"
- 0.8-0.6 – "God"
- 0.6-0.4 – "Moderat"
- 0.4-0.2 – "Dårlig"
- 0.2-0.0 – "Svært dårlig".

These categories indicate the ecosystem status. The nEQR categories were studied by creating a bar plot depicting the number of samples contained within each category of nEQR. This is displayed in *Figure 1*.

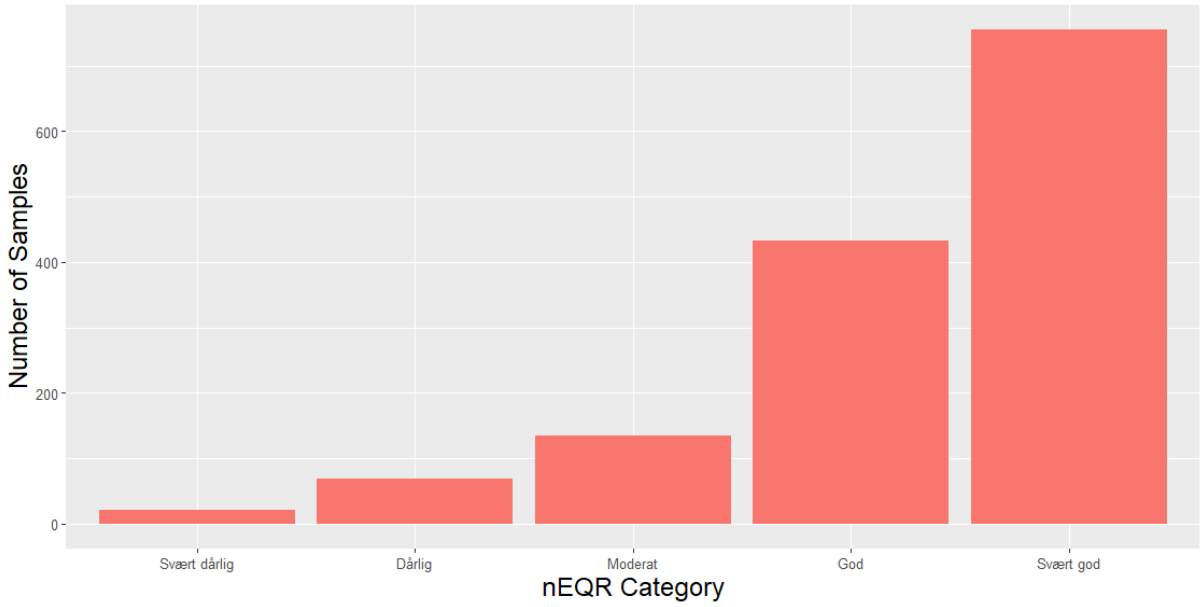

Figure 1. *A bar plot that shows the number of samples for each nEQR Category. The nEQR categories are "Svært god", which means nEQR values between 1.0-0.8, "God", which means nEQR values between 0.8-0.6, "Moderat", which means nEQR values between 0.6-0.4, "Dårlig", which means nEQR values between 0.4-0.2, and "Svært dårlig", which means nEQR values between 0.2-0.0.*

3.2 Taxonomic Predictors

3.2.1 VSEARCH vs DADA2

The two sets of taxonomic predictors (VSEARCH and DADA2) were compared to each other by making histogram plots showing the frequency of the library sizes, which refers to the total number of mapped reads, for each method. This is displayed in *Figure 2*. The number of predictors for VSEARCH was 43 767, but it was cut down to 28 548 after cutting out the prevalence of 1 percent. The number of predictors from DADA2 was 68 490, but it was cut down to 6875 after cutting out the prevalence of 1 percent.

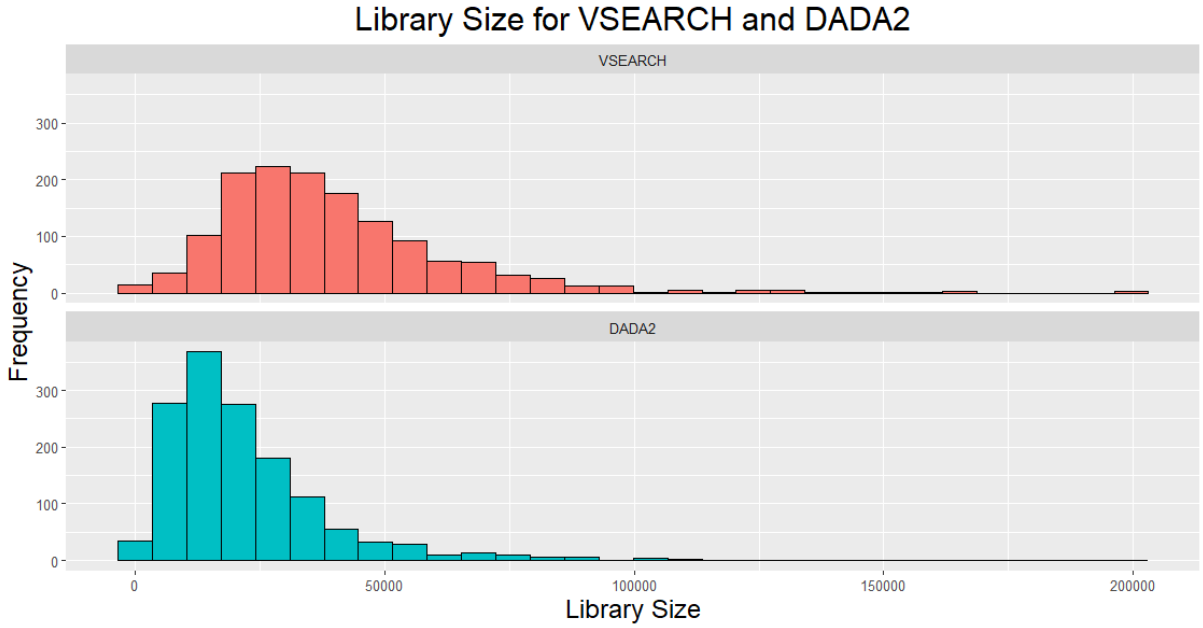

Figure 2. *Histograms that show the Library Size for VSEARCH (red) and DADA2 (blue). The x-axis shows the library size and y-axis shows the frequency of each library size.*

The taxonomic predictors from the VSEARCH and DADA2 methods were used to make predictions with PLS regression and LASSO regression. The taxa were also run with different models; one without normalization, one with TSS normalization and one with CLR normalization. These models were then compared to each other to determine the best combination of regression method and normalization model.

3.2.2 Normalization and Manhattan Distances

The best model for running the regression methods were determined by computing the average Manhattan distances for each component from each PLS regression object for the outputs from

both VSEARCH and DADA2. This is displayed in *Figure 3* for VSEARCH and in *Figure 4* for DADA2.

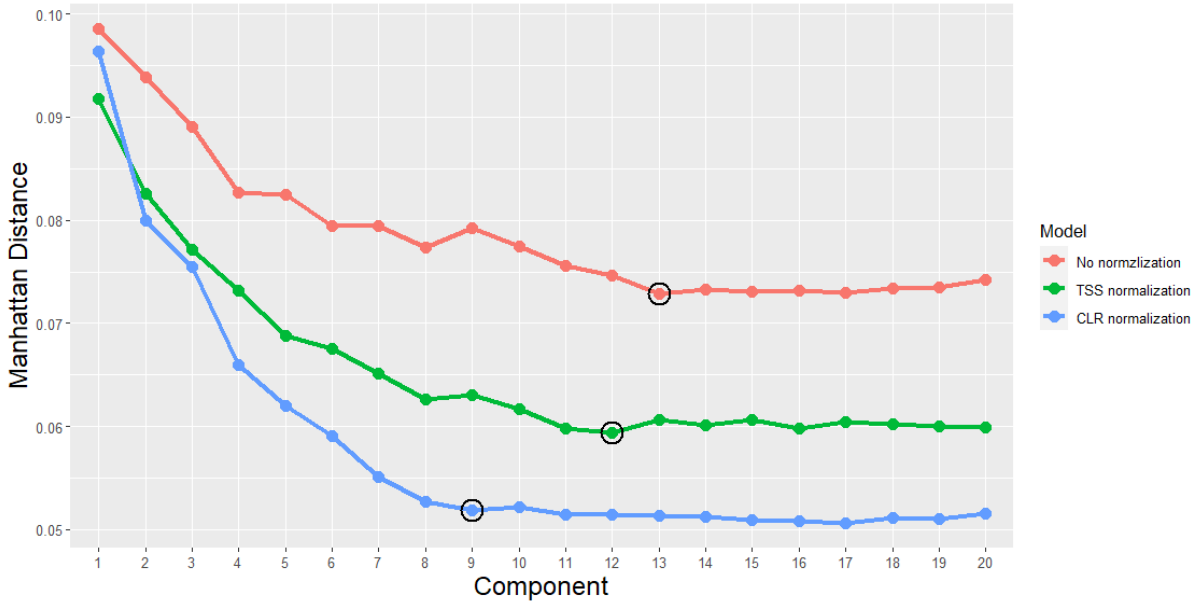

Figure 3. *Manhattan distances for the different models for VSEARCH with PLS regression. The x-axis shows each component for the PLS regression, which in this case was 20. The y-axis are the Manhattan distances for each component. The black circles mark the lowest Manhattan distances for each model, as well as their corresponding component.*

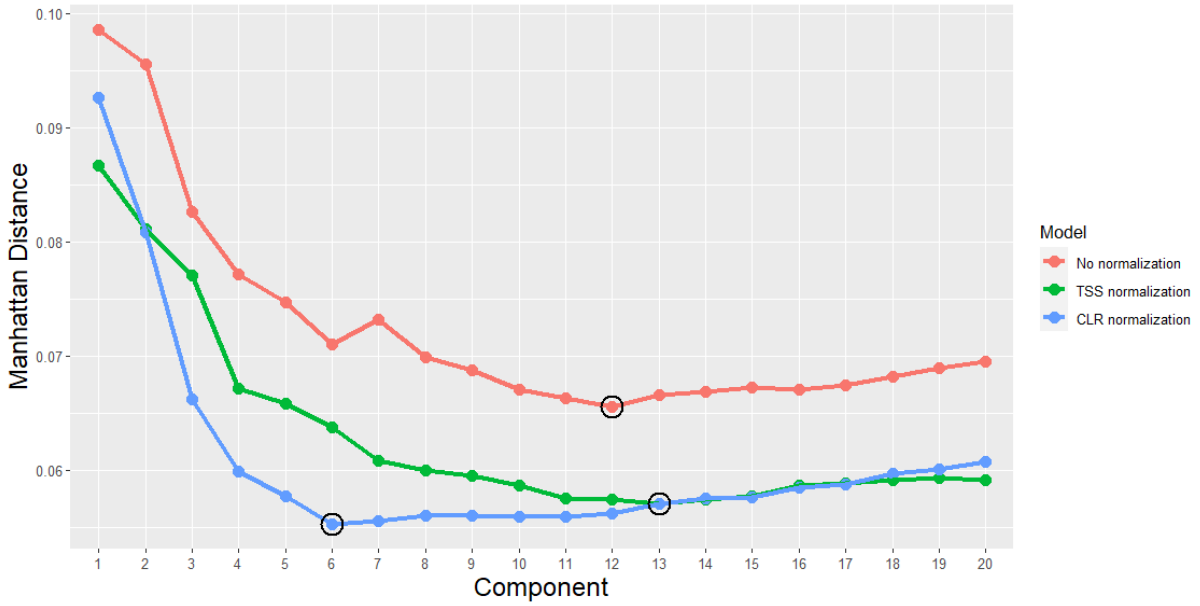

Figure 4. *Manhattan distances for the different models for DADA2 with PLS regression. The x-axis shows each component for the PLS regression, which in this case was 20. The y-axis are*

the Manhattan distances for each component. The black circles mark the lowest Manhattan distances for each model, as well as their corresponding component.

The plots above show that the CLR normalization is the best transformation for the outputs from the PLS regression for both VSEARCH and DADA2. The Manhattan distances for LASSO regression with CLR normalization were also computed, as well as the Manhattan distance for the PLS regression with the OTU's/ASV's from the LASSO regression for both VSEARCH and DADA2. All the Manhattan distances and their respective components for the different regression methods for VSEARCH and DADA2 with CLR normalization is displayed in *Table 2*.

Table 2. *This table shows the Manhattan distances and their respective components for the different regression methods for VSEARCH and DADA2 with CLR normalization. The LASSO regression methods show number of variables selected by LASSO.*

Method	**Manhattan Distance**	**Components/Variables from LASSO**
VSEARCH with PLS regression	0.052	9 (Components)
DADA2 with PLS regression	0.055	6 (Components)
VSEARCH with LASSO regression	0.033	410 (Variables)
DADA2 with LASSO regression	0.039	340 (Variables)
VSEARCH with PLS regression after LASSO regression	0.037	14 (Components)
DADA2 with PLS regression after LASSO regression	0.043	12 (Components)

The number of variables that are selected from LASSO can be shown by using the *plot()* function on the glmnet-object. This was done for both VSEARCH and DADA2, and these plots are shown in *Figure 5* for VSEARCH and *Figure 6* for DADA2.

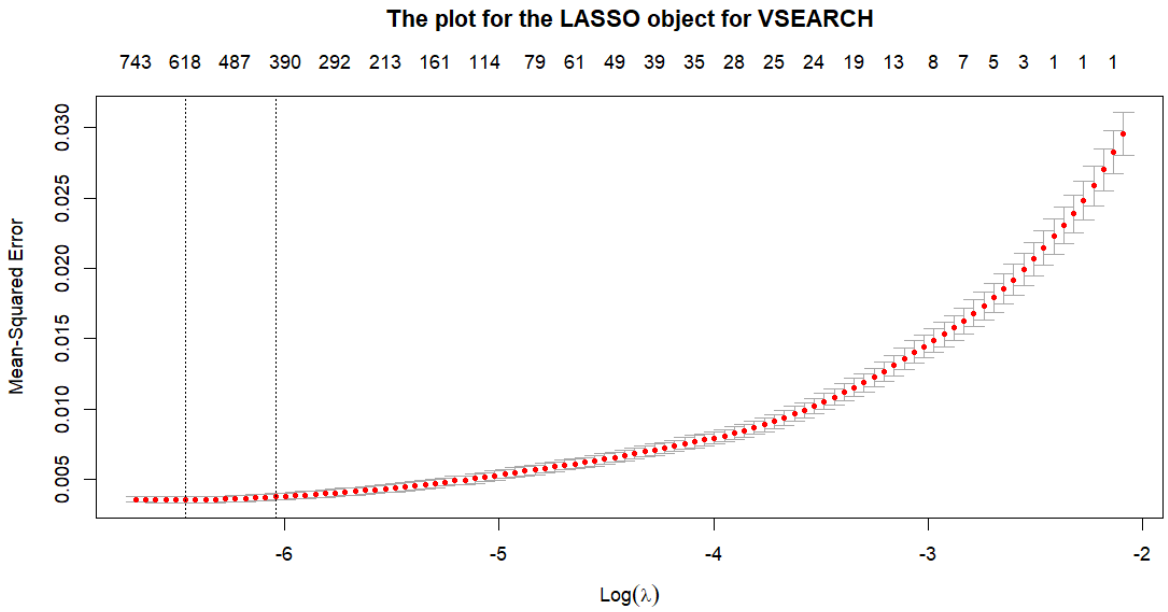

Figure 5. *This plot shows the LASSO object for VSEARCH.*

Figure 5 shows that the best variables for the LASSO object for VSEARCH is between 390 and 618. *Table 2* shows how many variables were actually chosen from LASSO and it was 410 variables, which is between 390 and 618.

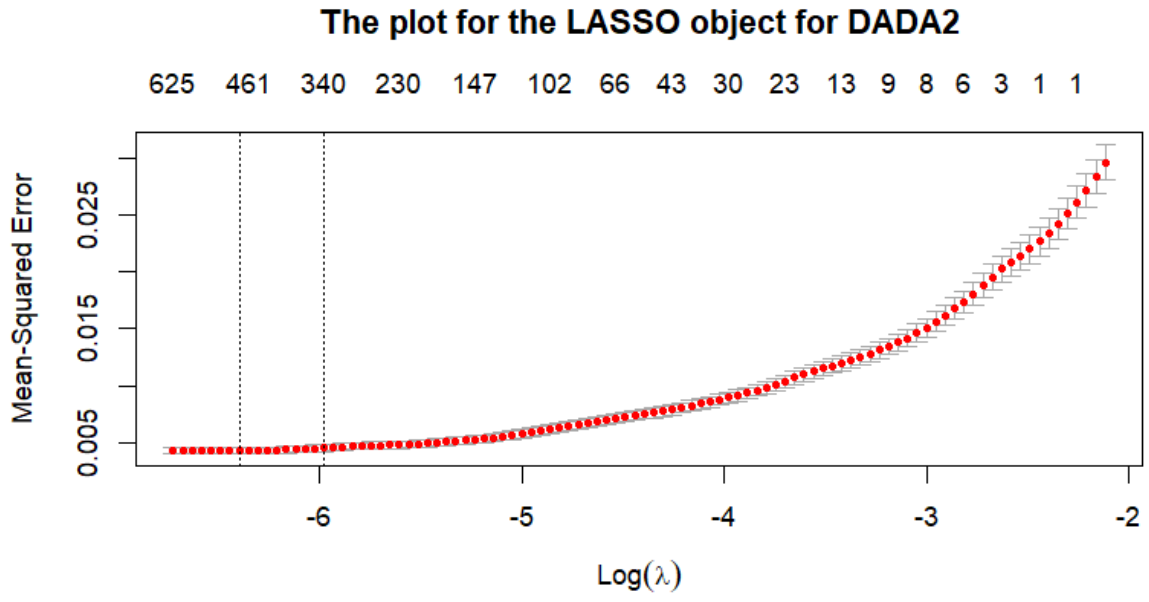

Figure 6. *This plot shows the LASSO object for DADA2.*

Figure 6 shows that the best variables for the LASSO object for DADA2 is between 340 and 461. *Table 2* shows how many variables were actually chosen from LASSO and it was 340 variables, which is between 340 and 461.

3.2.3 PLS Regression vs LASSO Regression

The CLR normalization was the best model for both VSEARCH and DADA2. Therefore, this model was used to make graphs for both PLS regression and LASSO regression, showing the observed nEQR values versus the predicted nEQR values. The PLS regression were also run on the OTU's chosen from the LASSO regression on VSEARCH, as well as the ASV's chosen from the LASSO regression for DADA2.

The PLS regression for VSEARCH and DADA2 are displayed in *Figure 7*.

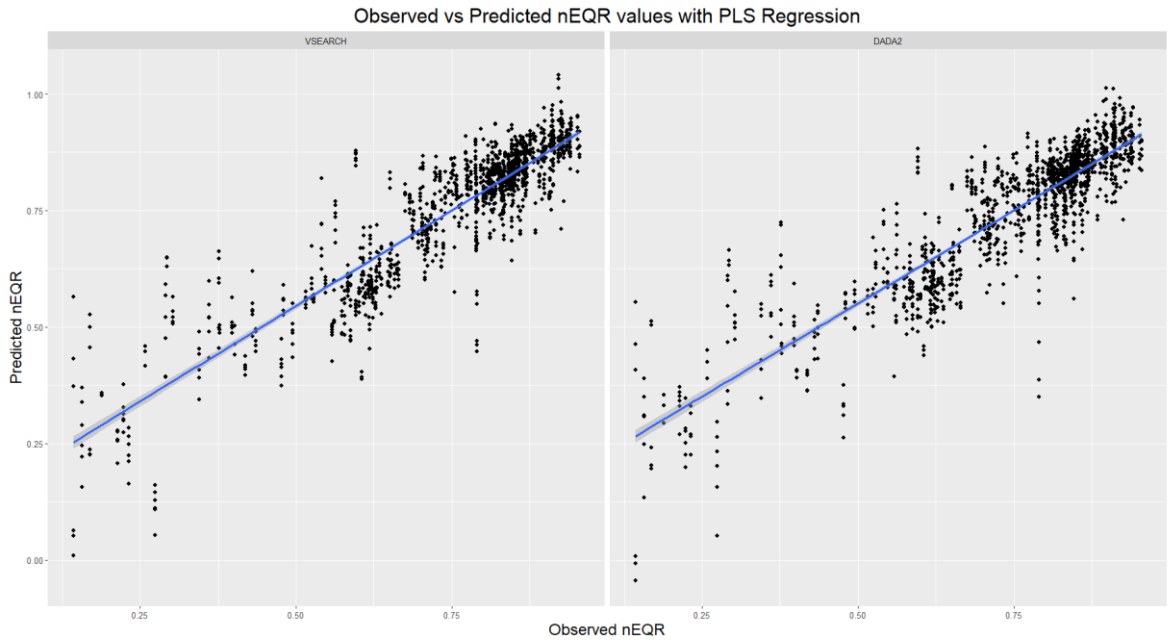

Figure 7. *PLS regression that shows the observed vs predicted nEQR values with CLR normalization for VSEARCH (left facet) and DADA2 (right facet). The x-axis is the observed nEQR values and the y-axis is the predicted nEQR values.*

The LASSO regression for VSEARCH and DADA2 are displayed in *Figure 8*.

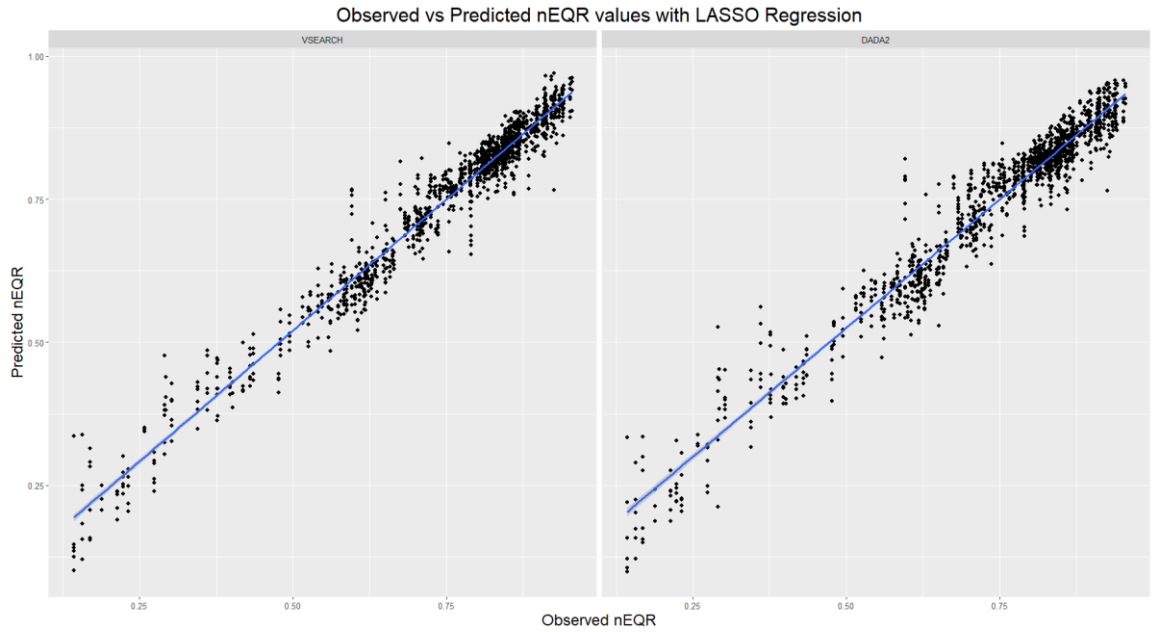

Figure 8. *LASSO regression that shows the observed vs predicted nEQR values with CLR normalization for VSEARCH (left facet) and DADA2(right facet). The x-axis is the observed nEQR values and the y-axis is the predicted nEQR values.*

The PLS regression on the OTU's/ASV's that were chosen from the LASSO regression are displayed in *Figure 9*.

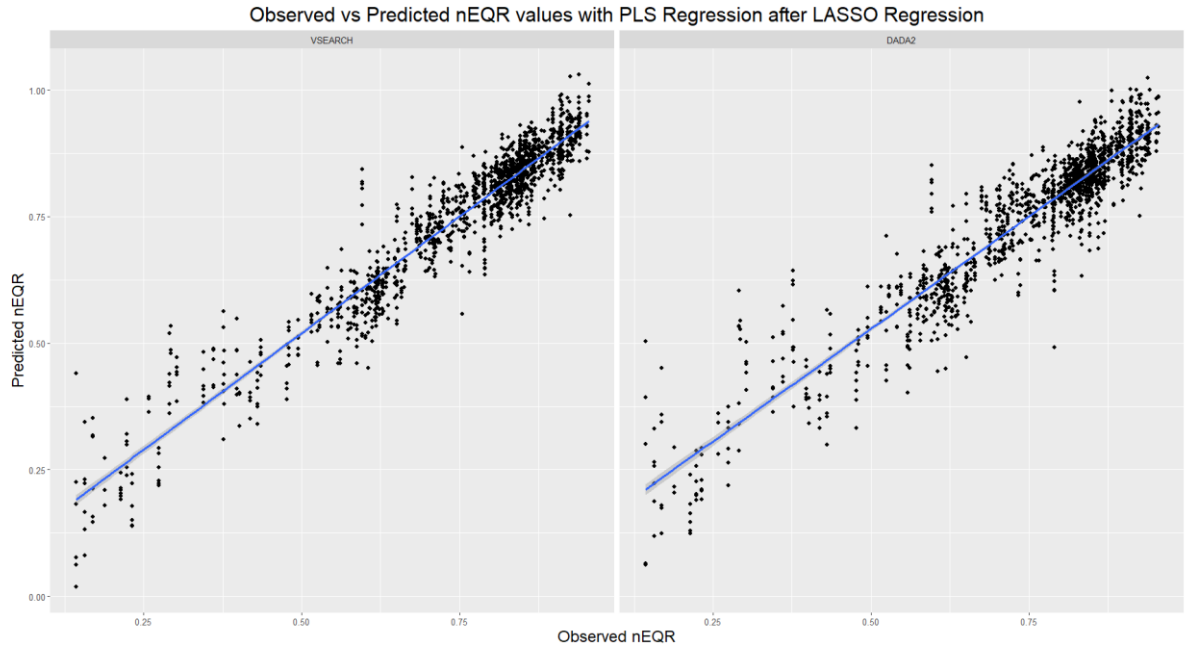

Figure 9. *PLS regression with the OTU's/ASV's from LASSO that shows the observed vs predicted nEQR values for VSEARCH (left facet) and DADA2 (right facet). The x-axis is the observed nEQR values and the y-axis is the predicted nEQR values.*

3.2.4 Taxa

The best method for predicting nEQR values were LASSO regression with CLR normalization for VSEARCH. This method was therefore used to investigate the taxonomy.

The nEQR values between 0.0 and 0.4 indicates pollution in the samples. To find out which genera causes pollution, a plot was made that shows the observed versus predicted nEQR values that are below 0.4. This plot included the three most abundant genera for each sample. This is shown in *Figure 10*.

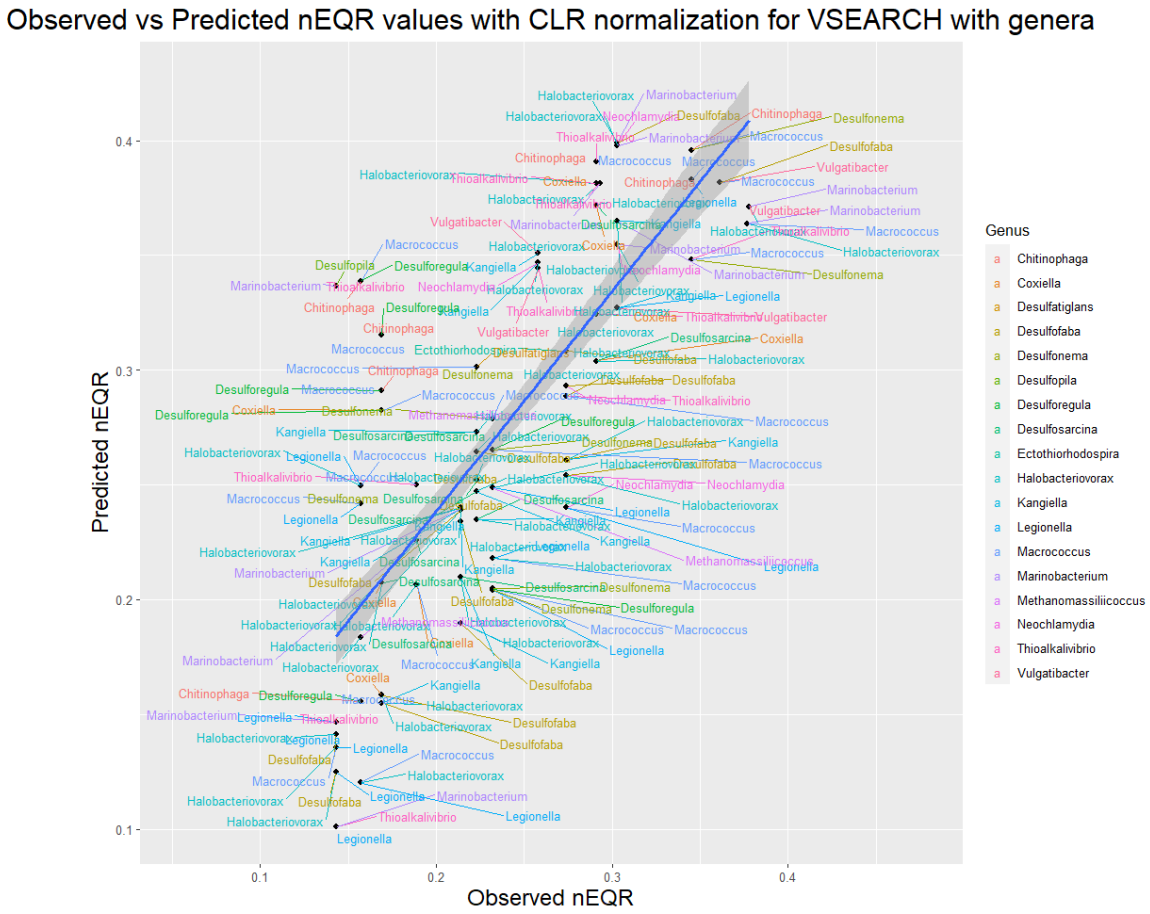

Figure 10. *The observed vs predicted nEQR values with CLR normalization for VSEARCH with the three most abundant genera for each sample with an nEQR value below 0.4. The x-axis is the observed nEQR values and the y-axis is the predicted nEQR values. The samples were coloured by genus.*

To make sure that the genera from *Figure 10* are more abundant for low nEQR values (0.0-0.4) compared to the rest of the samples, a Fisher Exact Test were performed on each genus from the figure. The p-value and odds ratio were retrieved from each of these tests and is shown in *Table 3*.

Table 3. *A table that shows the p-value and odds ratio from Fisher Exact Test for each genus in Figure 10.*

Genus Type	**p-value from Fisher Exact Test**	**Odds Ratio**
Chitinophaga	$2.749*10^{-6}$	18.32711
Coxiella	$9.217*10^{-5}$	7.000162
Desulfatiglans	0.5101	1.481749
Desulfofaba	$7.119*10^{-7}$	5.066082

Desulfonema	$2.339*10^{-8}$	36.85803
Desulfopila	0.1969	5.93687
Desulforegula	$1.106*10^{-9}$	Inf
Desulfosarcina	0.001444	3.276608
Ectothiorhodospira	0.9993	0.1321513
Halobacteriovorax	0.02538	1.45343
Kangiella	0.0004527	3.05104
Legionella	$7.54*10^{-7}$	6.291312
Macrococcus	$1.274*10^{-13}$	8.931768
Marinobacterium	0.003451	2.735989
Methanomassiliicoccus	0.001379	26.93902
Neochlamydia	0.04033	2.583946
Thioalkalivibrio	0.02269	2.14269
Vulgatibacter	1.0	0.1111317

The OTU's chosen from the LASSO regression were investigated by calculating the 10 highest and 10 lowest values of the regression coefficient multiplied by the standard deviation. These chosen OTU's and their values, as well as their corresponding genus were put in *Table 4*. Some of the genera were *NA's* and these were therefore removed from the table, and the OTU's with the highest and lowest values of the regression coefficient multiplied by the standard deviation that remained were respectively 8 and 9 OTU's.

Table 4. *A table that shows the 8 highest and 9 lowest values of the regression coefficient multiplied by the standard deviation and their corresponding OTU and Genus.*

	OTU	**Regression Coefficient * Standard Deviation**	**Genus**
Highest Values	OTU3	0.016396139	Actinopolymorpha
	OTU9	0.010750080	Rhodovibrio
	OTU191	0.009925091	Thiogranum
	OTU33	0.008210787	Filomicrobium
	OTU44814	0.008100746	Desulfovibrio
	OTU931	0.006558696	Rubritalea
	OTU150	0.006135673	Thioprofundum

	OTU45295	0.004202881	Sediminibacterium
Lowest Values	OTU10173	-0.007962028	Spiroplasma
	OTU393	-0.008724342	Spiroplasma
	OTU104	-0.008887960	Sulfurovum
	OTU224	-0.0091512960	Pelobacter
	OTU217	-0.009512960	Maribacter
	OTU43413	-0.009666421	Illumatobacter
	OTU46	-0.011733728	Desulfosarcina
	OTU1277	-0.022861025	Psychromonas
	OTU21	-0.026131834	Tetrasphaera

The number of genera that have NA values were calculated to be 116 out of 410.

3.3 Functional Predictors

The functional predictors from the outputs of PICRUSt2 were EC functions, KO functions and MetaCyc pathways.

3.3.1 PICRUSt2

The functional predictors from the PICRUSt2 method were used to make predictions with PLS regression and LASSO regression. The predictions were done using the outputs from VSEARCH. The functions were run with different models; one without normalization, one with TSS normalization and one with CLR normalization.

3.3.1.1 Normalization and Manhattan Distances

The best model for running the regression methods were determined by computing the average Manhattan distances for each component from each PLS regression object for the outputs from PICRUSt2. The Manhattan distances for the EC functions are displayed in *Figure 11*.

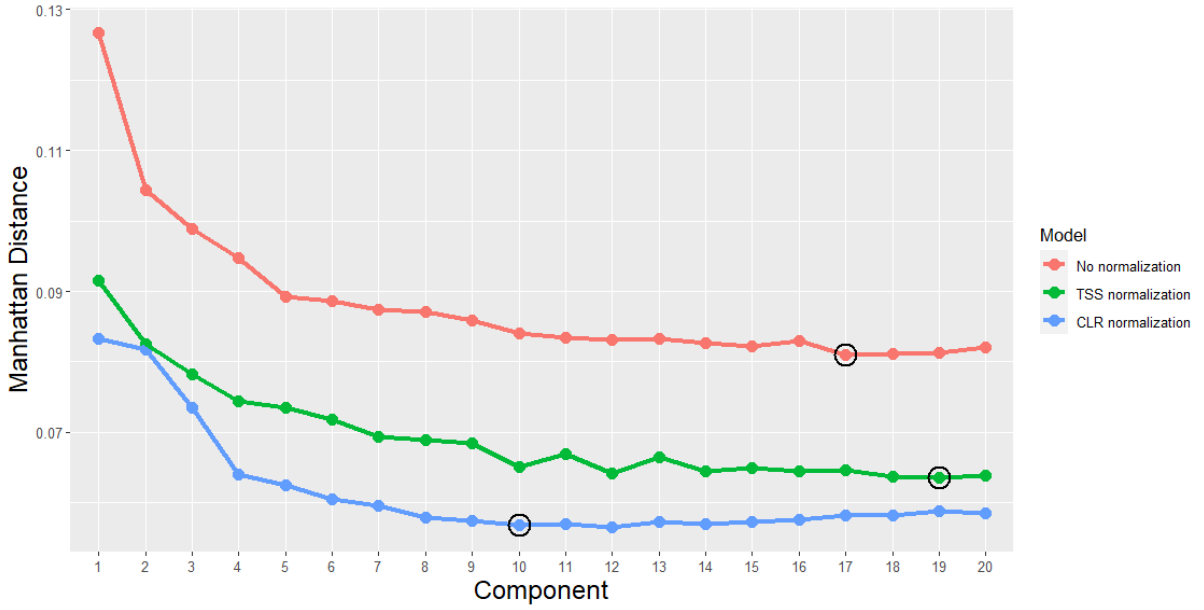

Figure 11. *Manhattan distances for the different models for EC functions from PICRUSt2 with PLS regression. The x-axis shows each component for the PLS regression, which in this case was 20. The y-axis are the Manhattan distances for each component. The black circles mark the lowest Manhattan distances for each model, as well as their corresponding component.*

The plot for Manhattan distances for EC functions shows that the CLR normalization is the most optimal. This was also the case for KO functions, as well as for MetaCyc pathways.

The Manhattan distances for the components from PLS regression with CLR normalization were identical for the EC functions, KO functions and MetaCyc pathways. All the Manhattan distances and their respective components for each method are displayed in *Table 5*.

Table 5. *This table shows the Manhattan distances for each method, as well as their respective component for the PLS regression. The LASSO regression methods show number of variables selected by LASSO.*

Method	**Manhattan Distance**	**Components/Variables from LASSO**
EC functions with PLS regression	0.057	10 (Components)
KO functions with PLS regression	0.057	10 (Components)
MetaCyc pathways with PLS regression	0.057	10 (Components)
EC functions with LASSO regression	0.046	129 (Variables)
KO functions with LASSO regression	0.045	119 (Variables)

MetaCyc pathways with LASSO regression	0.052	84 (Variables)
EC functions with PLS regression after LASSO regression	0.051	8 (Components)
KO functions with PLS regression after LASSO regression	0.048	8 (Components)
MetaCyc pathways with PLS regression after LASSO regression	0.057	9 (Components)

3.3.1.2 PLS Regression vs LASSO Regression

Since the Manhattan distances for the components from PLS regression with CLR normalization were identical for the EC functions, KO functions and MetaCyc pathways, the predicted nEQR values were the same for each of the three functional profiles. The plot that shows the observed versus predicted nEQR values for the PLS regression for each of the functional profiles is displayed in *Figure 12*.

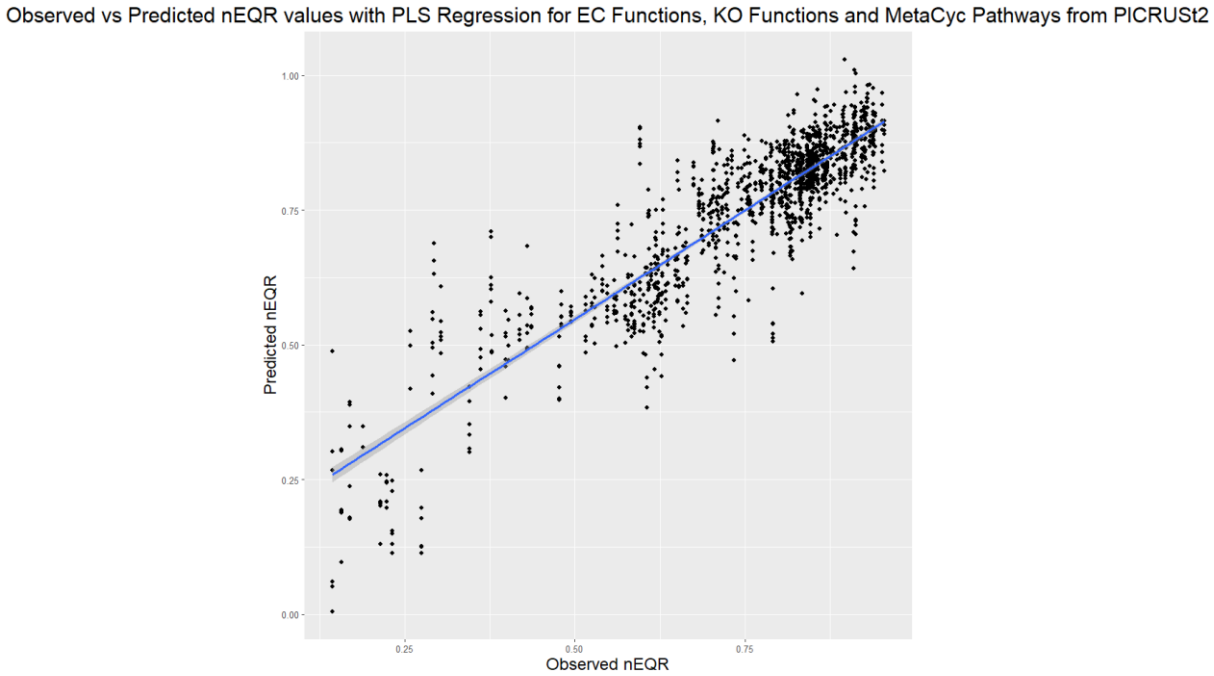

Figure 12. *PLS regression that shows the observed vs predicted nEQR values with CLR normalization for EC functions, KO functions and MetaCyc pathways from PICRUSt2. The x-axis is the observed nEQR values and the y-axis is the predicted nEQR values.*

The LASSO regression plots that shows the observed versus predicted nEQR values for EC functions, KO functions and MetaCyc pathways from PICRUSt2 are displayed in *Figure 13*.

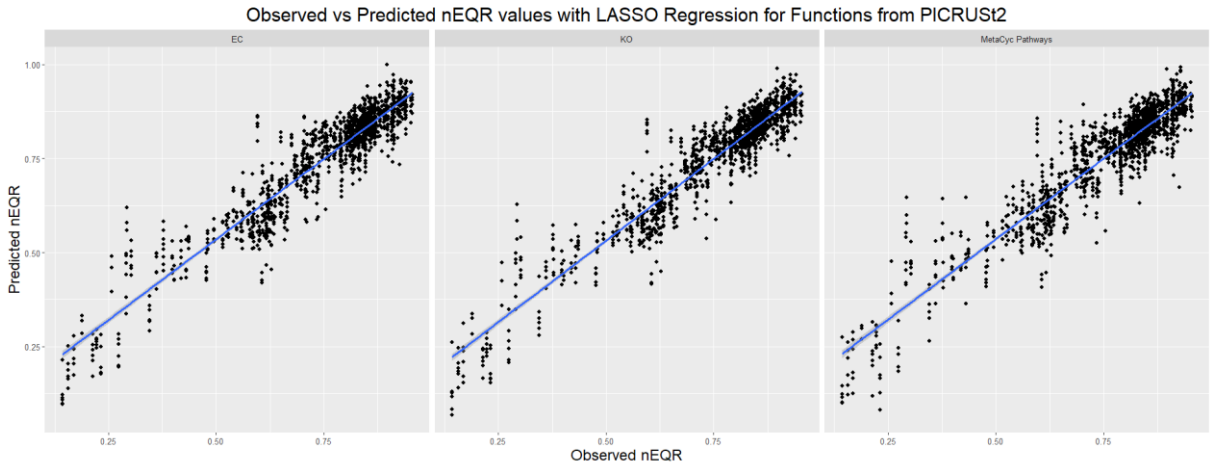

Figure 13. *LASSO regression that shows the observed vs predicted nEQR values with CLR normalization for EC functions (left facet), KO functions (middle facet) and MetaCyc Pathways (right facet) from PICRUSt2. The x-axis is the observed nEQR values and the y-axis is the predicted nEQR values.*

3.3.1.3 Functions

The EC functions, KO functions and MetaCyc pathways chosen from the LASSO regression were further investigated by calculating the 10 highest and 10 lowest values of the regression coefficient multiplied by the standard deviation.

20 of the EC functions chosen from the LASSO regression are displayed in *Table 6*

Table 6. *The 10 highest and 10 lowest values of the regression values of the regression coefficient multiplied by the standard deviation are displayed in this table, as well as their respective EC function and sysname.*

	Function	**Regression Coefficient × Standard Deviation**	**Sysname***
Highest Values	EC:2.6.1.96	0.024471769	4-aminobutanoate:pyruvate aminotransferase
	EC:2.1.1.157	0.019991626	S-adenosyl-L-methionine:sarcosine(or N,N-dimethylglycine) N-

			methyltransferase [N,N-dimethylglycine(or betaine)-forming]
	EC:2.7.14.1	0.019187822	ATP:[protein]-L-arginine Nomega-phosphotransferase
	EC:1.14.18.6	0.012547898	(4R)-4-hydroxysphinganine ceramide,ferrocytochrome-b5:oxygen oxidoreductase (fatty acyl 2-hydroxylating)
	EC:3.2.1.81	0.012171385	agarose 4-glycanohydrolase
	EC:1.1.1.11	0.011861593	D-arabinitol:NAD+ 4-oxidoreductase
	EC:1.3.3.11	0.010095005	6-(2-amino-2-carboxyethyl)-7,8-dioxo-1,2,3,4,7,8-hexahydroquinoline-2,4-dicarboxylate:oxygen oxidoreductase (cyclizing)
	EC:2.4.1.280	0.009552241	N,N'-diacetylchitobiose:phosphate N-acetyl-D-glucosaminyltransferase
	EC:2.8.3.1	0.009314867	acetyl-CoA:propanoate CoA-transferase
	EC:3.5.3.13	0.008484590	N-formimidoyl-L-glutamate iminohydrolase
Lowest Values	EC:4.1.1.86	-0.007645261	L-2,4-diaminobutanoate carboxy-lyase (propane-1,3-diamine-forming)
	EC:1.1.1.29	-0.007855915	D-glycerate:NAD+ oxidoreductase
	EC:3.6.3.31	-0.008028680	Hydrolases; Acting on acid anhydrides; Acting on acid anhydrides to catalyse transmembrane movement of substances
	EC:1.14.12.17	-0.008272201	nitric oxide,NAD(P)H:oxygen oxidoreductase
	EC:2.4.2.37	-0.009102146	NAD+:[dinitrogen reductase] (ADP-D-ribosyl)transferase

	EC:2.6.1.98	-0.010009663	UDP-2-acetamido-3-amino-2,3-dideoxy-alpha-D-glucuronate:2-oxoglutarate aminotransferase
	EC:3.1.2.1	-0.011304449	acetyl-CoA hydrolase
	EC:1.2.1.28	-0.012124914	benzaldehyde:NAD+ oxidoreductase
	EC:1.1.1.102	-0.015009937	D-erythro-dihydrosphingosine:NADP+ 3-oxidoreductase
	EC:1.2.1.9	-0.016176260	D-glyceraldehyde-3-phosphate:NADP+ oxidoreductase

* The names of these functions were found using the KEGG databases (Kanehisa and Goto, 2000).

20 of the KO functions chosen from the LASSO regression are displayed in *Table 7*.

Table 7. *The 10 highest and 10 lowest values of the regression values of the regression coefficient multiplied by the standard deviation are displayed in this table, as well as their respective KO function and name of the function.*

	Function	**Regression Coefficient × Standard Deviation**	**Name***
Highest Values	K01432	0.029722305	Arylformamidase [EC:3.5.1.9]
	K07275	0.014958659	outer membrane protein
	K06136	0.013487149	pyrroloquinoline quinone biosynthesis protein B
	K19586	0.012381190	membrane fusion protein, multidrug efflux system
	K10020	0.010026074	octopine/nopaline transport system permease protein
	K14762	0.008006707	ribosome-associated protein
	K18902	0.007211865	multidrug efflux pump
	K13041	0.006798988	two-component system, LuxR family, response regulator TtrR

	K11018	0.006764311	thermolabile hemolysin
	K14424	0.006405650	plant 4alpha-monomethylsterol monooxygenase [EC:1.14.18.11]
Lowest Values	K02075	-0.007177100	zinc/manganese transport system permease protein
	K19168	-0.007201294	toxin CptA
	K13378	-0.007518042	NADH-quinone oxidoreductase subunit C/D [EC:7.1.1.2]
	K10237	-0.008223554	trehalose/maltose transport system permease protein
	K08363	-0.008317438	mercuric ion transport protein
	K17752	-0.011464272	serine/threonine-protein kinase RsbT [EC:2.7.11.1]
	K04708	-0.013336519	3-dehydrosphinganine reductase [EC:1.1.1.102]
	K07234	-0.014459706	uncharacterized protein involved in response to NO
	K09688	-0.015267190	capsular polysaccharide transport system permease protein
	K09932	-0.016143576	uncharacterized protein

* The names of these functions were found using the KEGG databases (Kanehisa and Goto, 2000).

20 of the MetaCyc pathways chosen from the LASSO regression are displayed in *Table 8*.

Table 8. *The 10 highest and 10 lowest values of the regression values of the regression coefficient multiplied by the standard deviation are displayed in this table, as well as their respective MetaCyc pathway and the name of that pathway.*

	Pathway	**Regression Coefficient × Standard Deviation**	**Name***
Highest Values	P281-PWY	0.02578116	3-phenylpropanoate degradation
	PWY-5028	0.02442911	L-histidine degradation II
	PWY-6948	0.02107359	sitosterol degradation to androstenedione
	PWY-5419	0.02082307	catechol degradation to 2-hydroxypentadienoate II
	PWY-1882	0.01911994	superpathway of C1 compounds oxidation to CO_2
	PWY-1361	0.01901766	benzoyl-CoA degradation I (aerobic)
	AST-PWY	0.01644223	L-arginine degradation II (AST pathway)
	RHAMCAT-PWY	0.01596941	rhamnose catabolism rhamnose degradation
	PWY-6906	0.01543462	chitin derivatives degradation
	PWY-5910	0.01487468	superpathway of geranylgeranyldiphosphate biosynthesis I (via mevalonate)

Lowest Values	PWY-3661	-0.007500634	glycine betaine degradation I
	HSERMETANA-PWY	-0.007688220	
	PWY-5022	-0.009106710	4-aminobutanoate degradation V
	PWY-6174	-0.009237970	mevalonate pathway II (haloarchaea)
	PWY-6396	-0.009381117	superpathway of 2,3-butanediol biosynthesis
	DENITRIFICATION-PWY	-0.015001891	nitrate reduction I (denitrification)
	PWY-5705	-0.015302831	allantoin degradation to glyoxylate III
	PYRIDOXSYN-PWY	-0.015371944	Pathway: pyridoxal 5'-phosphate biosynthesis I
	P124-PWY	-0.020719525	Bifidobacterium shunt
	3-HYDROXYPHENYLACETATE-DEGRADATION-PWY	-0.027343986	4-hydroxyphenylacetate degradation

* The names of these pathways were found using the MetaCyc database (Caspi, et.al., 2010).

3.3.1.4 Functions compared to each other

The different functional profiles from PICRUSt2, EC functions, KO functions and MetaCyc pathways, were compared to each other by performing hierarchical clustering with average linkage. This was done to find out which functional profile was the best, or if they were similar to each other. The hierarchical clustering tree were cut into 16, 32 and 64 clusters, and then the rand index for each cluster between each functional profile were compared to each other. This is displayed in *Table 9*.

Table 9. *This table shows the rand index between each functional profile with 16, 32 and 64 number of groups.*

Function	**Rand Index**	**The number of groups (k)**
EC functions – KO functions	0.81	16
EC functions – MetaCyc pathways	0.63	16
KO functions – MetaCyc pathways	0.52	16
EC functions – KO functions	0.74	32
EC functions – MetaCyc pathways	0.82	32
KO functions – MetaCyc pathways	0.78	32
EC functions – KO functions	0.82	64
EC functions – MetaCyc pathways	0.86	64
KO functions – MetaCyc pathways	0.87	64

3.4 Taxonomic Predictors vs Functional Predictors

The Manhattan distances for the components from PLS regression with CLR normalization for the taxonomic predictors and functional predictors were compared to each other by making a line chart plot. In this plot, the taxonomic predictors are from VSEARCH. Since components from the CLR normalization for all the functional predictors were identical, the EC functions, KO functions and MetaCyc pathways are represented by one line in the plot. This plot is displayed in *Figure 14*.

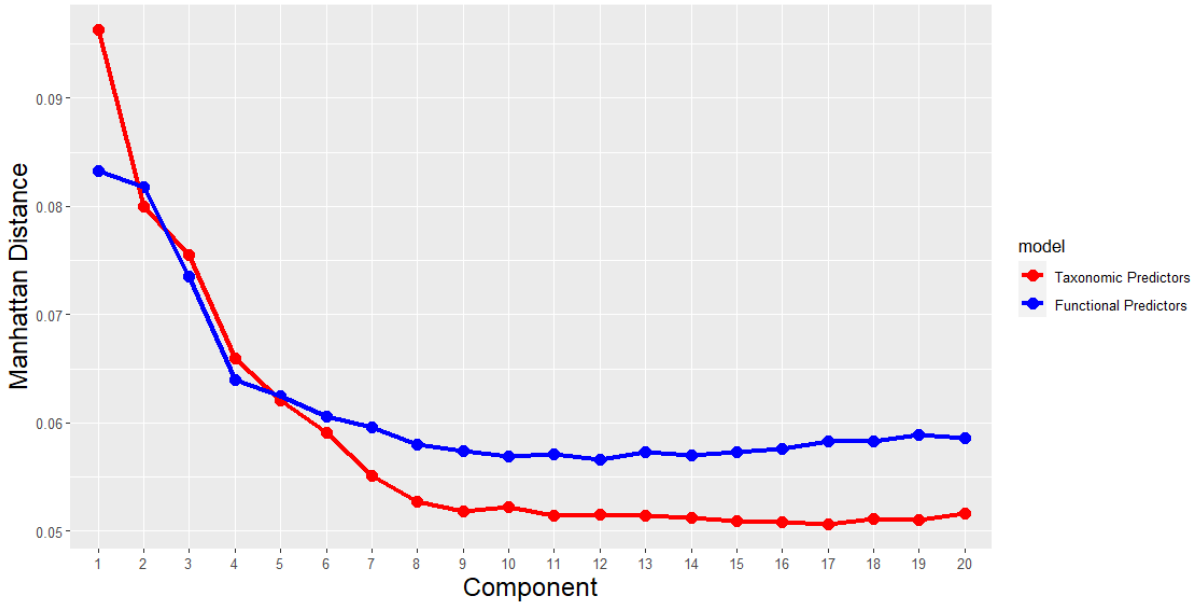

Figure 14. *Manhattan distances for VSEARCH (red) and PICRUSt2 (blue). The x-axis shows each component for the PLS regression, which in this case was 20. The y-axis are the Manhattan distances for each component.*

3.5 Overall results

The minimum Manhattan distance with CLR normalization for both taxonomic (VSEARCH) and functional predictors, and their corresponding components are displayed in *Table 10*.

Table 10. *This table shows the minimum Manhattan distance with CLR normalization from PLS regression for both taxonomic (VSEARCH) and functional (EC, KO, MetaCyc) predictors, as well as their corresponding components.*

Method	**Manhattan Distance**	**Components**
Taxa (VSEARCH)	0.052	9
EC functions	0.057	10
KO functions	0.057	10
MetaCyC pathways	0.057	10

The best method for predicting nEQR values for both taxonomic and functional predictors were LASSO regression with CLR normalization. The Manhattan distances for these methods are displayed in *Table 11*.

Table 11. *This table shows the minimum Manhattan distance with CLR normalization from LASSO regression for both taxonomic (VSEARCH) and functional (EC, KO, MetaCyc) predictors, as well as their corresponding components.*

Method	**Manhattan Distance**
Taxa (VSEARCH)	0.033
EC functions	0.046
KO functions	0.045
MetaCyC pathways	0.052

The LASSO regression will extract a certain number of variables from the readcount tables for both the taxonomic and functional predictors. The numbers of these variables are displayed in *Table 12*.

Table 12. *This table shows the number of variables chosen from the LASSO regression for each method.*

Method	**Number of Variables from LASSO**
Taxa (VSEARCH)	410 / 28 548
EC functions	129 / 2329
KO functions	119 / 7634
MetaCyc functions	84 / 430

The methods used for predictions in this thesis are taxonomic and functional predictors. The taxonomic predictors used to analyse which method is the best based on prediction errors (the difference between observed and predicted nEQR values), were the predictions from the VSEARCH tool.

Therefore, the methods compared to each other are the taxonomic predictors from VSEARCH, as well as the functional predictors from the EC functions, KO functions and the MetaCyc pathways. These methods were compared to each other by making a box plot that shows each prediction method on the x-axis and the absolute prediction errors on the y-axis. This is displayed in *Figure 15* for the predictions from the PLS regressions and in *Figure 16* for the predictions from the LASSO regressions.

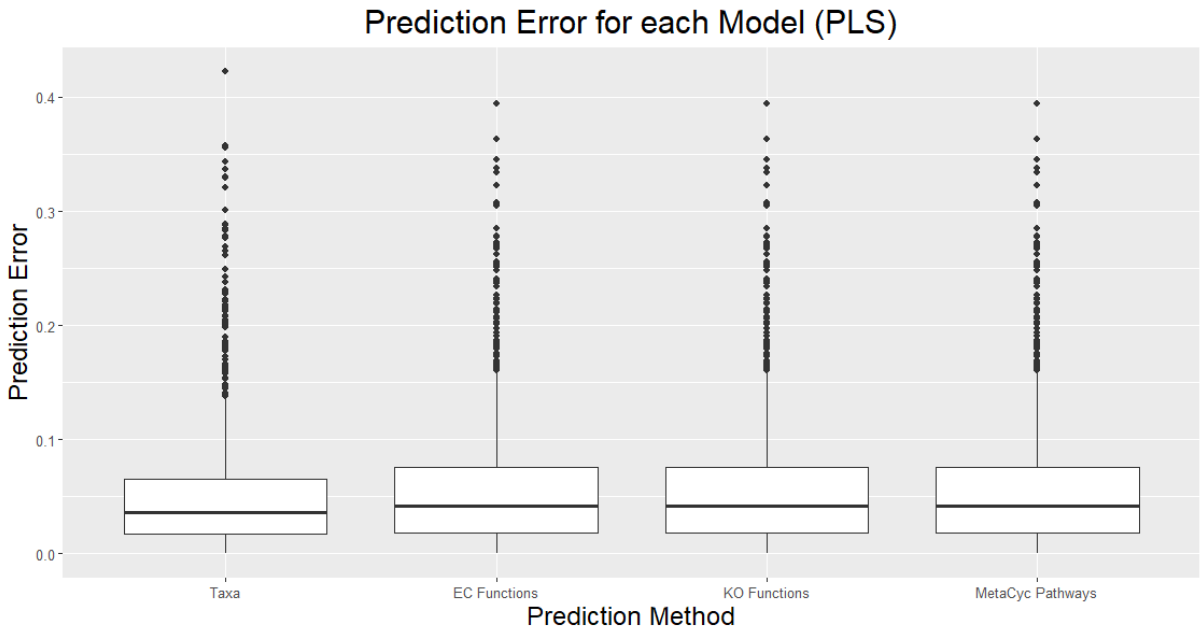

Figure 15. *A box plot that shows the absolute prediction error for each model with the predictions from the PLS regressions. The x-axis shows the prediction method, and the y-axis shows the absolute prediction error.*

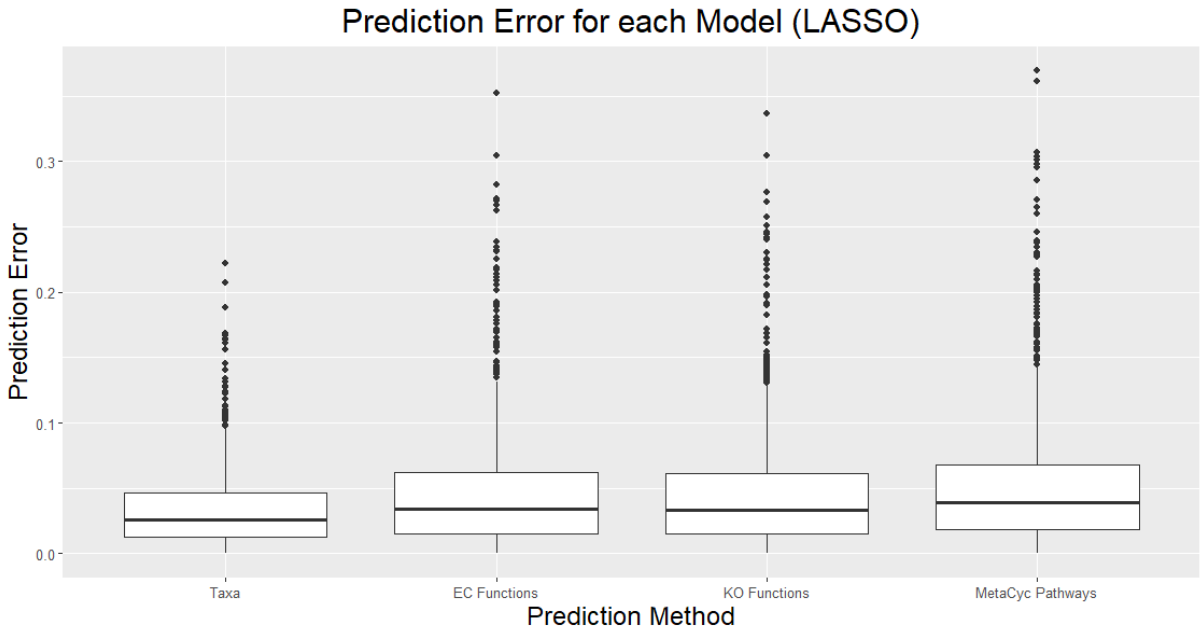

Figure 16. *A box plot that shows the absolute prediction error for each model with the predictions from the LASSO regressions. The x-axis shows the prediction method, and the y-axis shows the absolute prediction error.*

The prediction error for each nEQR category was also compared to each other. This was done by making a box plot that shows each nEQR category on the x-axis and the prediction errors on the y-axis. This is displayed in *Figure 17* for the predictions from the PLS regressions and in *Figure 18* for the predictions from the LASSO regressions.

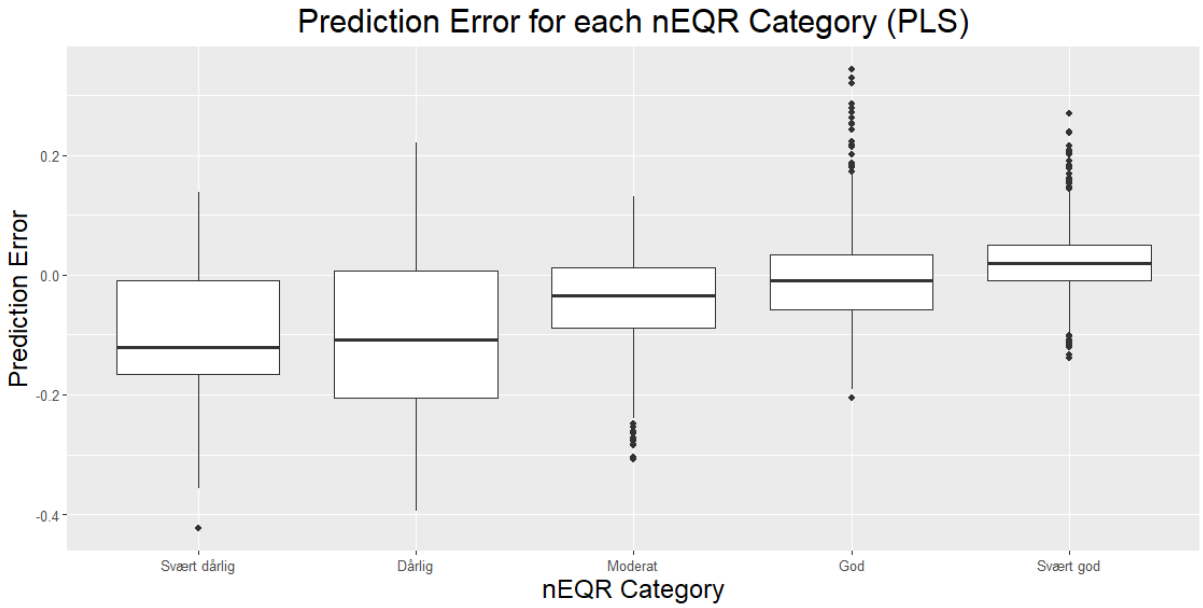

Figure 17. *A box plot that shows the prediction error with the predictions from the PLS regressions for each nEQR category. The x-axis shows the nEQR category, and the y-axis shows the prediction error.*

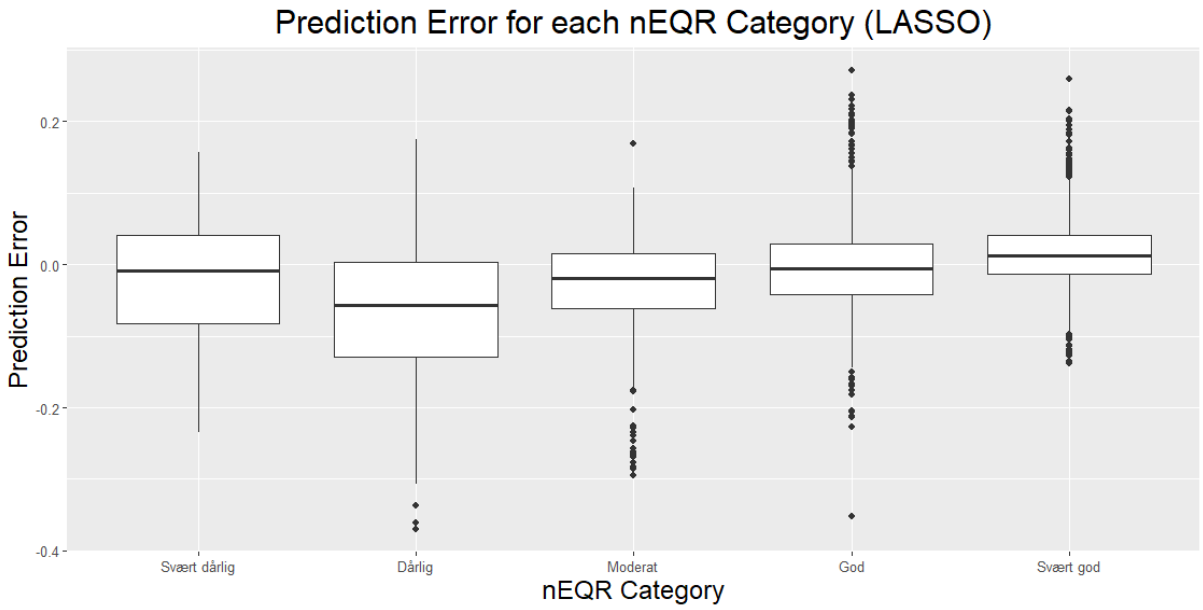

Figure 18. *A box plot that shows the prediction error with the predictions from the LASSO regressions for each nEQR category. The x-axis shows the nEQR category, and the y-axis shows the prediction error.*

4. Discussion

4.1 The Data

The main aim of this thesis was to predict the nEQR values based on taxonomic profiling. The two methods used for taxonomic profiling was VSEARCH and DADA2. Predicting the nEQR values was done using two different machine learning methods: PLS regression and LASSO regression. In addition to taxonomic profiling, the nEQR values were also predicted using functional profiling, and the aim of this was to investigate whether functional profiling could provide anything beyond taxonomic profiling.

The data used in this thesis was from the AQUAeD project and it contained 1414 samples with nEQR values (without NA values). The nEQR values were divided into 5 categories, which indicated the ecosystem status: 1.0-0.8 – "Svært god", 0.8-0.6 – "God", 0.6-0.4 – "Moderat", 0.4-0.2 – "Dårlig" and 0.2-0.0 – "Svært dårlig". The bar plot in *Figure 1* showed the distribution of the different nEQR categories. This bar plot showed that the two last categories, which indicates good ecosystem, had far more samples than the two first categories, which indicates a polluted ecosystem. The number of samples for this project is large, but the variations of the nEQR categories are significant with more samples with higher nEQR values. This can mean that the machine learning methods are better at predicting the higher nEQR values compared to the lower values. The plots that were made to predict the observed and predicted nEQR values (*Figure 7, 8, 9, 12 and 13*) also reflects this.

4.2 Taxonomic Predictors

4.2.1 VSEARCH vs DADA2

The two sets of taxonomic predictors (VSEARCH and DADA2) were compared to each other by making histogram plots showing the frequency of the library sizes, which refers to the total number of mapped reads, for each method. This was displayed in *Figure 2*, and it shows that in general the library sizes for VSEARCH are larger than for DADA2, but one library size is larger for DADA2 than for VSEARCH. The number of predictors for VSEARCH was 43 767, but it was cut down to 28 548 after cutting out the prevalence of 1 percent. The number of predictors from DADA2 was 68 490, but it was cut down to 6875 after cutting out the prevalence of 1 percent.

The taxonomic predictors from the VSEARCH and DADA2 methods were used to make predictions with PLS regression and LASSO regression. The regression methods were run with different models; one without normalization, one with TSS normalization and one with CLR normalization. These models were then compared to each other to determine the best combination of regression method and normalization model.

4.2.2 Normalization and Manhattan Distances

To find the best combination of regression method and normalization model, the average Manhattan distances, i.e. the differences between observed and predicted nEQR, were computed. This was done for each component from each PLS regression object for the outputs from both VSEARCH and DADA2, and it was displayed in *Figure 3* for VSEARCH and *Figure 4* for DADA2. These plots showed that the CLR normalization was the most optimal model for both VSEARCH and DADA2. The Manhattan distances were then computed for the outputs from the LASSO regression with CLR normalization, as well as the Manhattan distances for the PLS regression with the OTU's/ASV's from the LASSO regression for both VSEARCH and DADA2. All the Manhattan distances and their corresponding components for the different regression methods with CLR normalization were displayed in *Table 2*. From this table, it is clear that the best combination of regression method and normalization model is LASSO regression with CLR normalization for VSEARCH since this combination had the lowest Manhattan distance with 0.033. *Figure 5* and *6* shows how many variables that LASSO picked out from VSEARCH and DADA2 respectively. This means that the number of OTU's picked from the LASSO object for VSEARCH was between 390 and 616 out of 43 767, and the actual number was 410 OTU's, which is between these numbers. The number of ASV's chosen from the LASSO object from the LASSO object for DADA2 was between 340 and 461 out of 6875, and the actual number was 340, which is between these numbers.

4.2.3 PLS Regression vs LASSO Regression

The two regression methods, PLS regression and LASSO regression, with CLR normalization were further investigated by making plots that showed the observed nEQR values on the x-axis versus the predicted nEQR values on the y-axis. These plots are displayed in *Figure 7* for PLS regression, *Figure 8* for LASSO regression and in *Figure 9* for PLS regression after LASSO regression. These plots further confirms that the best combination is the LASSO regression with CLR normalization for VSEARCH. This combination is shown in *Figure 8* on the left facet.

All of these figures, show that both regression methods are better at predicting the higher nEQR values compared to the lower nEQR values, which is expected since there are more samples with a higher nEQR value. The boxplots in *Figure 17* and *18* also reflects this. These figures shows that the categories "Svært god", "God" and "Moderat" is better at predicting the nEQR values compared to the categories "Dårlig" and "Svært dårlig", which is to be expected as there are far more number of samples in the three first categories, compared to the to last.

4.2.4 Taxa

The taxonomy from VSEARCH was further investigated, and the best method for predicting nEQR values were LASSO regression with CLR normalization for VSEARCH. The nEQR values between 0.0 and 0.4 indicated pollution in the samples. To find out which genera indicates pollution, a plot was made that shows the observed versus predicted nEQR values that are below 0.4. This plot included the three most abundant genera for each sample. This is shown in *Figure 10*. This plot shows the taxonomy that indicates pollution and the taxa from this plot was further investigated in *Table 3*. This table shows the genus type with their corresponding p-value and odds ratio from Fisher Exact Test.

The odds ratio (OR) measures the strength and direction of the association between two categorical categories. If the odds ratio is greater than 1, it indicates that the event, i.e. genera that causes pollution, is more likely to occur in the presence of the first variable compared to its absence. Conversely, an odds ratio less than 1 suggests that the event is less likely to occur in the presence of the first variable.

The genera with p-values below 0.05 and odds ratio over 1 is *Chitinophaga*, *Coxiella*, *Desulfofaba*, *Desulfonema*, *Desulforegula*, *Desulfosarcina*, *Halobacteriovorax*, *Kangiella*, *Legionella*, *Macrococcus*, *Marinobacterium*, *Methanomassiliicoccus*, *Neochlamydia* and *Thioalkalivibrio*.

The taxonomy was also investigated by looking at the OTU's chosen from the LASSO regression. These OTU's were chosen by calculating the 10 highest and 10 lowest values of the regression coefficient multiplied by the standard deviation. These chosen OTU's and their values, as well as their corresponding genus were put in *Table 4*. Some of the genera were *NA's* and these were therefore removed from the table, and the OTU's with the highest and lowest values of the regression coefficient multiplied by the standard deviation that remained were respectively 8 and 9 OTU's. This table (*Table 4*) shows the genera for the highest and lowest values of the regression coefficient multiplied by the standard deviation. The genera from this

are *Actinopolymorpha*, *Rhodovibrio*, *Thiogranum*, *Filomicrobium*, *Desulfovibrio*, *Rubritalea*, *Thioprofundum* and Sediminibacterium for the highest values, and *Spiroplasma*, *Spiroplasma*, *Sulfurovum*, *Pelobacter*, *Maribacter*, *Illumatobacter*, *Desulfosarcina*, *Psychromonas* and *Tetrasphaera*.

From the different methods of retrieving the bacteria that causes pollution, a common denominator are sulfuric bacteria. This means that sulfuric bacterium is one of the bacteria that indicate pollution in fish farms. Sulfate is highly abundant in marine settings, and it is the most prevalent driver of the respiration of organic matter under anoxic conditions. Dissimilatory sulfate reduction yields hydrogen sulfide, which poses toxicity to the aerobic respiratory pathway (Flood, et., al., 2021).

4.3 Functional Predictors

The functional predictors from the outputs of PICRUSt2 were EC functions, KO functions and MetaCyc pathways.

4.3.1 PICRUSt2

The functional predictors from the PICRUSt2 method were used to make predictions with PLS regression and LASSO regression. The predictions were done using the outputs from VSEARCH. The functions were run with different models; one without normalization, one with TSS normalization and one with CLR normalization. These models were then compared to each other to determine the best combination of regression method and normalization model.

4.3.1.1 Normalization and Manhattan

To find the best combination of regression method and normalization model, the average Manhattan distances, i.e. the differences between observed and predicted nEQR, were computed. This was done for each component from each PLS regression object for the outputs from EC functions, KO functions and MetaCyc pathways from PICRUSt2, and it was displayed in *Figure 11* for EC functions, and it showed that the CLR normalization was the most optimal model the EC functions from PICRUSt2. This was also proved to be true for the KO functions, as well as the MetaCyc pathways. The Manhattan distances were then computed for the outputs from the LASSO regression with CLR normalization, as well as the Manhattan distances for the PLS regression with the OTU's/ASV's from the LASSO regression from all functions from PICRUSt2. All the Manhattan distances and their corresponding components for the different regression methods with CLR normalization were displayed in *Table 5*. From this table, it is

clear that the combinations of regression method and normalization model are very similar to each other. The different regression methods with CLR normalization were further investigated by making plots that showed the observed nEQR values on the x-axis versus the predicted nEQR values on the y-axis.

4.3.1.2 PLS Regression vs LASSO Regression

The two regression methods, PLS regression and LASSO regression, with CLR normalization were further investigated by making plots that showed the observed nEQR values on the x-axis versus the predicted nEQR values on the y-axis. These plots are displayed in *Figure 12* for the PLS regressions (these plots were the same for each function from PICRUSt2, therefore it is the same plot for each PLS regression for each PICRUSt2 function), and *Figure 13* for the LASSO regression for the EC functions, KO functions and MetaCyc pathways. These plots further confirms that the best combination is the LASSO regression with CLR normalization for the KO functions. This combination is shown in *Figure 13* in the middle facet.

All of these figures, show that both regression methods for each function from PICRUSt2 are better at predicting the higher nEQR values compared to the lower nEQR values.

4.3.1.3 Functions

The functions from PICRUSt2 were further investigated. The functions from PICRUSt2 were EC functions, KO functions and MetaCyc pathways, and these functions from the LASSO regression were further investigated by calculating the 10 highest and 10 lowest values of the regression coefficient multiplied by the standard deviation.

20 of the EC functions chosen from the LASSO regression were displayed in *Table 6*, 20 of the KO functions chosen from the LASSO regression were displayed in *Table 7* and 20 of the MetaCyc pathways chosen from the LASSO regression were displayed in *Table 8*. More information about the EC functions and KO functions can be found in the KEGG databases, and more information about the MetaCyc pathways can be found in the MetaCyc database.

4.3.1.4 Functions compared to each other

The different functional profiles from PICRUSt2, EC functions, KO functions and MetaCyc pathways, were compared to each other by performing hierarchical clustering with average linkage. This was done in order to verify whether the functions are similar to each other or not, since the results are very similar to each other. Therefore, the question would be whether it is really necessary to have three different functions from PICRUSt2, or if one of them would have

been enough. The hierarchical clustering tree were cut into 16, 32 and 64 clusters, and then the rand index for each cluster between each functional profile were compared to each other. This was displayed in *Table 9*. The Rand index is a way to compare the similarity of results between two different clustering methods. The Rand index always takes on a value between 0 and 1 where:

- **0:** Indicates that two clustering methods do not agree on the clustering of any pair of any elements.
- **1:** Indicates that two clustering methods perfectly agree on the clustering of every pair of elements.

(Bobbitt, 2021)

All of the Rand indexes were above 0.5, which means that all the clustering between functional profiles is good. However, some of these Rand indexes are above 0.8, which can be seen in *Table 9*, which means that the clustering between these functional profiles is almost perfectly agreeable. If the Rand Index is close to 1, then the results between two different clustering methods are almost identical. This means that the clustering between EC and KO functions with 16 number of groups are almost identical. The clustering between EC and KO functions with 32 groups are almost identical. The clustering between EC functions and MetaCyc pathways with 32 groups are almost identical. The clustering between KO functions and MetaCyc pathways with 32 groups are almost identical. The clustering between EC and KO functions with 64 groups are almost identical. The clustering between EC functions and MetaCyc pathways with 64 groups are almost identical. Lastly, the clustering between KO functions and MetaCyc pathways with 64 groups are almost identical (see *Table 9*).

4.4 Taxonomic Predictors vs Functional Predictors

The taxonomic and functional predictors were compared to each other in a number of ways. First of all, the Manhattan distances with CLR normalization were compared to each other. This was done for both PLS regression and LASSO regression. The plot in *Figure 14* shows the Manhattan distances for each component of the PLS regression for both VSEARCH and PICRUSt2. This displays that the Manhattan distances is initially larger for the taxonomic predictors for the first components, but by the sixth component and onwards, the Manhattan distances are lower for the taxonomic predictors compared to the functional predictors. The minimum Manhattan distance with CLR normalization for the PLS regression for taxonomic

and functional predictors, as well as their corresponding components are displayed in *Table 9*. This shows that the best predictors are taxonomic.

4.5 Overall Results

The nEQR values were also predicted using LASSO regression with CLR normalization for both taxonomic and functional predictors, and the Manhattan distances for these methods were displayed in *Table 10*. This table also shows that the taxonomic predictors are the best.

The LASSO regression extracted a certain number of variables from the readcount tables for both the taxonomic and functional predictors. The number of these variables were displayed in *Table 11*. This also displays that the best predictors are the taxonomic from VSEARCH.

The taxonomic and functional predictors were further investigated by making box plots that shows the prediction error for each model (Taxa, EC functions, KO functions and MetaCyc pathways). This is displayed in *Figure 15* for PLS regression and *Figure 16* for LASSO regression. Both of these figures shows that the taxonomic predictors are slightly better than the functional predictors.

The prediction error was aslo computed for each nEQR category for both PLS and LASSO regression. These boxplots are displayed in *Figure 17* and *18*, and they show that the better the nEQR category, the better the prediction error.

4.6 Concluding remarks and further perspective

The aim of this study was to predict the nEQR values based foremostly on taxonomic profiling. The taxonomic predictors were produced using two different methods, VSEARCH and DADA2. The nEQR values were then predicted using two different machine learning methods, PLS regression and LASSO regression. The best method for taxonomic profiling was VSEARCH, and therefore the outputs from this method was used to predict the nEQR values using functional profiling using PICRUSt2. This was done to see if the functional predictors could provide something more than the taxonomic predictors.

The taxonomic and functional predictors were compared to each other with Manhattan distances and prediction errors, which were displayed in figures and tables. The results from this shows that the taxonomic predictors were slightly better for both Manhattan distances and prediction errors. This confirms that the functional predictors do not provide anything beyond the taxonomic predictions. This means that the nEQR values could be predicted solely based on

taxonomic profiling. This could solely be based on the results from PICRUSt2. There is other software that can extract the functional predictors based on the taxonomic profiling. These methods could perhaps be better at predicting the functional profiles compared to PICRUSt2, but for this thesis there was not enough time.

However, the data contributed for this thesis was unequally distributed, which is not ideal when running machine learning methods. Most of the samples had an nEQR value above 0.6, which categorises as good or very good. The samples with nEQR values below 0.4 are categorized as bad or very bad, and there were very few samples below this value. This means that the machine learning methods will be better at predicting higher nEQR values compared to the lower values. This is also reflected in the figures (*Figure 7, 8, 9, 12 and 13*) that show the predictions for the nEQR values.

The fact that most of the samples have a high nEQR value is good for the fish farms, since it means that they are healthy and does not need to be quarantined, but it is not ideal for an experiment like this.

The ideal would be to have the same number of samples for each nEQR category since this would give better predictions for all nEQR values, and therefore better results overall.

In conclusion, predicting the nEQR values using taxonomic and functional predictors with PLS regression and LASSO regression could be a great tool in the fish farming industry in the future. The best prediction of PLS and LASSO regression is the LASSO regression with VSEARCH and CLR normalization. The Manhattan distance for this is 0.33, which is a third of the error of the predictions. This means that the predictions are good. Besides, the higher nEQR values are better to predict compared to the lower values, but it is the lower values that are of most interest because it is these values that determine whether a quarantine period should be imposed on the fish farm.

The conclusion of this thesis is that it is possible to predict the nEQR values using taxonomic and functional predictors with machine learning methods, such as PLS regression and LASSO regression. But more research should be done so that the predictions are as accurate as they can be when this is implemented on fish farms.

Bibliography

Abdi, H. (s.a.). *Partial Least Squares (PLS) Regression.* The University of Texas at Dallas. Available at: https://personal.utdallas.edu/~herve/Abdi-PLS-pretty.pdf

Allen, M. P. (1997). *Understanding Regression Analysis.* Available at: file:///C:/Users/Caroline/Downloads/b102242.pdf

Allon, G., Chen, D., Jiang, Z. and Zhang, D. J. (s.a.). *Machine Learning and Prediction Errors in Casual Inference.* Available at: file:///C:/Users/Caroline/Downloads/SSRN-id4480696.pdf

Barbera, P., Kozlov, A., M., Czech, L., Morel, B., Darriba, D., Flouri, T. and Stamatakis, A. (2018). EPA_ng: Massively Parallel Evolutionary Placement of Genetic Sequences. *Systematic Biology.* Volume 68, Issue 2, Pages 365-369. https://doi.org/10.1093/sysbio/syy054

Bioinformatic Methods for Biodiversity Metabarcoding. (s.a.). *2. (not filtering) Dereplication.* Available at: https://learnmetabarcoding.github.io/LearnMetabarcoding/filtering/not_filtering_dereplication.html

Bobbitt, Z. (2021). *What is the Rand Index? (Definition & Examples).* Available at: https://www.statology.org/rand-index/

Callahan, B. J., McMurdie, P. J., Rosen, M. J., Han, A. W. Johnson, A. J. A. and Holmes, S. P. (2016). DADA2: High resolution sample inference from Illumina amplicon data. PMCID: PMC4927377; NIHMSID: NIHMS782534; PMID: 27214047, doi: 10.1038/nmeth.3869

Callahan, B. J., McMurdie, P. J., Rosen, M., J., Han, A., W., Johnson, A., J., A. and Holmes, S., P. (2016). DADA2: High resolution sample inference from Illumina amplicon data. PMCID: PMC4927377; NIHMSID: NIHMS782534; PMID: 27214047. doi: 10.1038/nmeth.3869

Caspi R., Altman T., Dale J. M., Dreher K., Fulcher C. A., Gilham F., Kaipa P., Karthikeyan A. S., Kothari A., Krummenacker M., Latendresse M., Mueller L. A., Paley S., Popescu L., Pujar A., Shearer A. G., Zhang P., Karp P. D. (2010). The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases. Nucleic Acids Res. PMID: 19850718; PMCID: PMC2808959. doi: 10.1093/nar/gkp875

Caspi, R., Billington, R., Keseler, I., M., Kothari, A., Krummenacker, M., Midford, P., E., Ong, W., K., Paley, S., Subhraveti, P. and Karp, P. D. (2020). The MetaCyc database of metabolic pathways and enzymes – a 2019 update. PMID: 31586394; PMCID: PMC6943030. doi: 10.1093/nar/gkz862

Creative Biolabs. (s.a.). *Community Analysis Using Next-generation Sequencing.* Available at: https://live-biotherapeutic.creative-biolabs.com/community-analysis-using-next-generation-sequencing.htm?gclid=CjwKCAiA1-6sBhAoEiwArqlGPmO9Cc9YEJX9ONadqFdD36-7CftU6rEtgx5ACfU5cJtmoghw_tT5KhoC0rIQAvD_BwE

Czech, L., Barbera, P. and Stamatakis, A. (2020). Genesis and Gappa: processing, analyzing and visualizing phylogenetic (placement) data. *Bioinformatics,* Volume 36, Issue 10, Pages 3263-3265. https://doi.org/10.1093/bioinformatics/btaa070

DADA2, (s.a.). *DADA2 Pipeline Tutorial (1.16).* Available at: https://benjjneb.github.io/dada2/tutorial.html

DADA2, (s.a.). *DADA2: Fast and accurate sample inference from amplicon data with single-nucleotide resolution.* Available at: https://benjjneb.github.io/dada2/

Danicic, A., Vucic, N., Kasapovic, S. and Pajic, V. (2018). *Taxonomic Profiling of Metagenomics Samples.* Available at: https://www.sevenbridges.com/taxonomic-profiling-of-metagenomics-samples/

Douglas, G. (2021). *Picrust2. Full pipeline script.* Available at: https://github.com/picrust/picrust2/wiki/Full-pipeline-script

Douglas, G. (2021). *Picrust2. Home.* Available at: https://github.com/picrust/picrust2/wiki#citations

Douglas, G. (2023). *Picrust2. PICRUSt2 Tutorial (v2.5.2).* Available at: https://github.com/picrust/picrust2/wiki/PICRUSt2-Tutorial-(v2.5.2)

Douglas, G., M., Maffei, V., J., Zaneveld, J. R., Yurgel, S., N., Brown, J., R., Taylor, C., M., Hutttenhower, C. and Langille, M., G., I. (2020). PICRUSt2 for prediction of metagenome functions. PMCID: PMC7365738; NIHMSID: NIHMS1602025; PMID: 32483366. doi: 10.1038/s41587-020-0548-6

Douglas, G., M., Maffei, V., J., Zaneveld, J. R., Yurgel, S., N., Brown, J., R., Taylor, C., M., Hutttenhower, C. and Langille, M., G., I. (2020). PICRUSt2 for prediction of metagenome functions. *Nat Biotechnol* **38**, 685–688. https://doi.org/10.1038/s41587-020-0548-6

Flood, B. E., Louw, D., C., Van der Plas, A., K. and Bailey, J., V. (2021). Giant sulfur bacteria (Beggiatoaceae) from sediments underlying the Benguela upwelling system host diverse microbiomes. PMCID: PMC8612568 | PMID: 34818329, doi: 10.1371/journal.pone.0258124

Franzosa, E. A., McIver, L. J., Rahnavard, G., Thompson, L. R., Schirmer, M., Weingart, G., Lipson, K., S., Knight, R., Caporaso, J., G., Segata, N. and Huttenhower, C. (2018). Species-level functional profiling of metagenomes and metatranscriptomes. PMCID: PMC6235447; NIHMSID: NIHMS1507068; PMID: 30377376, doi: 10.1038/s41592-018-0176-y

Friedman, J., Hastie, T., Tibshirani, R., Narasimhan, B., Tay, K., Simon, N., Qian, J. and Yanf, J. (2023). glmnet: Lasso and Elastic-Net Regularized Generalized Linear Models. Journal of statistical Software. https://CRAN.R-project.org/package=glmnet

GeeksforGeeks. (2023). *Cross Validation in Machine Learning.* Available at: https://www.geeksforgeeks.org/cross-validation-machine-learning/

Gloor, G., B., Macklaim, J., M., Pawlowsky-Glahn, V. and Egozcue, J., J. (2017). Microbiome Datasets Are Compositional: And This Is Not Optional. PMCID: PMC5695134; PMID: 29187837; doi: 10.3389/fmicb.2017.02224

Hastie, T., Tibshirani, R. and Friedman, J. (2008). *The Elements of Statistical Learning. Data Mining, Inference, and Prediction.* 2. Edition. Springer. Available at: https://eduumb-my.sharepoint.com/personal/lars_snipen_nmbu_no/Documents/undervisning/emner/STAT340/2023/ESLII.pdf

HMMER. (s.a.). *HMMER: biosequence analysis using profile hidden Markov models.* Available at: http://hmmer.org/

Hu, Q. N., Zhu, H., Li, X., Zhang, M., Deng, Z., Yang, X. and Deng, Z. (2012). Assignment of EC Numbers to Enzymatic Reactions with Reaction Difference Fingerprints. PMCID: PMC3532301; PMID: 23285222, doi: 10.1371/journal.pone.0052901

Illumina. (s.a.). *An Effective Biomonitoring Tool.* Available at: https://www.illumina.com/techniques/sequencing/dna-sequencing/targeted-resequencing/environmental-dna.html

Illumina. (s.a.). *Overview of Environmental Metagenomics.* Available at: https://www.illumina.com/areas-of-interest/microbiology/environmental-metagenomics.html

Illumina. (s.a.). *What Are 16S and ITS rRNA Sequencing?*. Available at: https://www.illumina.com/areas-of-interest/microbiology/microbial-sequencing-methods/16s-rrna-sequencing.html

James, G., Witten, D., Hastie, T. and Tibshirani, R. (2021). *An Introduction to Statistical Learning with Applications in R.* Second Edition. Available at: file:///C:/Users/Caroline/AppData/Local/Microsoft/Windows/INetCache/Content.Outlook/76BWXONM/ISLRv2_website.pdf

Janda, J. M. and Abbott S. L., (2007). 16S rRNA Gene Sequencing for Bacterial Identification in the Diagnostic Laboratory: Pluses, Perils, and Pitfalls. PMCID: PMC2045242. doi: 10.1128/JCM.01228-07

Kanehisa, M., Furumichi, M., Tanabe, M., Sato, Y. and Morishima, K. (2017). KEGG: new perspectives of genomes, pathways, diseases and drugs. PMCID: PMC5210567; PMID: 27899662, doi: 10.1093/nar/gkw1092

Kanehisa, M. and Goto, S. (2000). KEGG: kyoto encyclopedia of genes and genomes. Nucleic Acid Res. PMID: 10592173; PMCID: PMC102409. doi: 10.1093/nar/28.1.27

Karp, P. D., Riley, M., Paley, S. M. and Pellegrini-Toole, A. (2002). The MetaCyc Database. PMCID: PMC99148; PMID: 11752254. doi: 10.1093/nar/30.1.59

KEGG Enzyme Database. (s.a.). Available at: https://www.genome.jp/dbget-bin/www_bfind?enzyme

KEGG Orthology Database. (s.a). Available at: https://www.genome.jp/dbget-bin/www_bfind?orthology

KO (KEGG ORTHOLOGY) Database. (2023). *KO Database of Molecular Functions.* Available at: https://www.genome.jp/kegg/ko.html

Kumar, D. (2023). *A Complete understanding of LASSO Regression.* Available at: https://www.mygreatlearning.com/blog/understanding-of-lasso-regression/

Kyoto University. (1995). *KEGG (Kyoto Encyclopedia of Genes and Genomes* [database]. Retrieved from: https://www.kegg.jp/

Langille, M., Zaneveld, J., Caporaso, J., McDonald, D., Knights, D., Reyes, J., Clemente, J. Burkepile, D., Thurber, R., Knight, R., Beiko, R. and Huttenhower, C. (2013). Predictive functional profiling of microbial communities using 16S rRNA marker gene sequences. *Nat Biotechnol* **31**, 814–821. https://doi.org/10.1038/nbt.2676

Liland, K. H., Mevik, B. H., Wehrens, R. and Hiemstra, P. (2023). pls: Partial Least Squares and Principal Component Regression. https://CRAN.R-project.org/package=pls

Louca, S. and Doebeli, M. (2017). Efficient comparative phylogenetics on large trees. *Bioinformatics,* Volume 34, Issue 6, Pages 1053-1055. https://doi.org/10.1093/bioinformatics/btx701

MetaCyc. (s.a.). *MetaCyc Metabolic Pathway Database.* Available at: https://metacyc.org/

Mirarab, S., Nguyen, N. and Warnow, T. (2011). SEPP: SATé-Enabled Phylogenetic Placement. *Biocomputing 2012,* pp. 247-258. https://doi.org/10.1142/9789814366496_0024

National Library of Medicine (s.a.). *SRA.* Available at: https://www.ncbi.nlm.nih.gov/sra

PennState Eberly College of Science. (2018). *Lesson 9: RNA Seq Data.* The Pennsylvania State University. Available at: https://online.stat.psu.edu/stat555/node/13/

Pérez-Cobas, A. E., Gomez-Valero, L. and Buchrieser, C. (2020). Metagenomic approaches in microbial ecology: an update on whole-genome and marker gene sequencing analyses. PMCID: PMC7641418. doi: 10.1099/mgen.0.000409

Pirouz, D., M., (2006). *An Overview of Partial Least Squares.* University of Western Ontario – The Richard Ivey School of Business. Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1631359

R Development Core Team. (2010). R: A language and environment for statistical computing. In R Foundation for Statistical Computing. http://www.R-project.org

Reading Council of Norway. (s.a.). *KSP: On-Site monitoring of aquaculture impact on the environment by open-source nanopore eDNA analysis.* Available at: https://prosjektbanken.forskningsradet.no/project/FORISS/320076?Kilde=FORISS&distribution=Ar&chart=bar&calcType=funding&Sprak=no&sortBy=date&sortOrder=desc&resultCount=30&offset=0&TemaEmne.2=Jord

Rognes T, Flouri T, Nichols B, Quince C, Mahé F. (2016) VSEARCH: a versatile open source tool for metagenomics. PeerJ 4:e2584. doi: 10.7717/peerj.2584

Rognes T, Flouri T, Nichols B, Quince C, Mahé F. (2016). VSEARCH: a versatile open source tool for metagenomics. PeerJ., PMID: 27781170; PMCID: PMC5075697., doi: 10.7717/peerj.2584.

Rolling, T., Zhai, B., Frame, J., Hohl, T. M., and Taur, Y. (2022). Customization of a *DADA2*-based pipeline for fungal transcribed spacer 1 (ITS1) amplicon data sets. PMCID: PMC8765055. doi: 10.1172/jci.insight.151663

Ruscheweyh, H. J., Milanese, A., Paoli, L., Karcher, N., Clayssen, Q., Keller, M. I., Wirbel, J., Bork, P., Mende, D. R., Zeller, G. and Sunagawa, S. (2022). Cultivation-independent genomes greatly expand taxonomic-profiling capabilities of mOTUs across various environments. *Microbiome* **10**, 212. https://doi.org/10.1186/s40168-022-01410-z

Snipen, L. (2023). *Bin310 module 10 – Metabarcoding data.* Norwegian University of Life Sciences. Available at: https://arken.nmbu.no/~larssn/teach/bin310/module_10_metabarcoding_processing.html#5_The_vsearch_read_processing

SRI International. (1997). *MetaCyc* [database]. Retrieved from: https://metacyc.org/

StatsTest. (2024). *Fisher's Exact Test.* Available at: https://www.statstest.com/fischers-exact-test/

Tucci, L. (2023). *What is machine learning and how does it work? In-depth guide.* Available at: https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML

Vannportalen. (2018). *Klassifiseringsveileder 02:2018 Klassifisering av miljøtilstand i vann. Økologisk og kjemisk klassifiseringssystem for kystvann, grunnvann, innsjøer og elver.* Available at: https://www.vannportalen.no/veiledere/klassifiseringsveileder/

Vinje, H. and Snipen, L. (2023). *BIN310 – module 11.* Norwegian University of Life Sciences. Available at: https://arken.nmbu.no/~larssn/teach/bin310/module_11_metabarcoding_analysis.html#41_The_TSS

Wakefield, K. (s.a.). *A guide to the types of machine learning algorithms and their applications.* Available at: https://www.sas.com/en_gb/insights/articles/analytics/machine-learning-algorithms.html

Wickham, H. (2016). ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York. https://ggplot2.tidyverse.org

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., ... & Yutani, H. (2019). Welcome to the tidyverse. Journal of Open Source Software, 4(43), 1686. https://doi.org/10.21105/joss.01686

Xia, Y. (2023). Statistical normalization methods in microbiome data with application to microbiome research. PMCID: PMC10461514; PMID: 37622724, doi: 10.1080/19490976.2023.2244139

Ye, Y. and Doak, T., G. (2009). A Parsimony Approach to Biological Pathway Reconstruction/Inference for Genomes and Metagenomes. PLoS Comput Biol 5(8): e1000465. https://doi.org/10.1371/journal.pcbi.1000465

Zhang L., Chen F., Zeng Z., Xu M., Sun F., Yang L., Bi X., Lin Y., Gao Y., Hao H., Yi W., Li M. and Xie Y. (2021). Advances in Metagenomics and Its Application in Environmental Microorganisms. *Front. Microbiol.* 12:766364. doi: 10.3389/fmicb.2021.766364

Zhang, C., Chen, Z., Zhang, M. and Jia, S., (2023). KEGG_Extractor: An Effective Extraction Tool for KEGG Orthologs. PMCID: PMC9956942; PMID: 36833314; doi: 10.3390/genes14020386

Norges miljø- og biovitenskapelige universitet
Noregs miljø- og biovitskapelege universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway