The Development of a Functional Annotation Pipeline to Characterise Metagenome-Assembled Genomes of Microorganisms Found in Anaerobic Digestion

Louise Erica Grimble

MSc by Research

University of York
Biology

February 2023

Abstract

Anaerobic digestion involves the conversion of organic waste into biogas and biofertilisers. Anaerobic digesters are commonly found within the wastewater treatment process in the UK, converting waste sludge into methane. Higher yields of methane are required for AD to become a favourable renewable energy source. The AD process consists of four steps (hydrolysis, acidogenesis, acetogenesis, and methanogenesis) that are driven by complex microbial communities. Hydrogenotrophs and methanogens are rate-determining factors, highlighting the significance of these microbial communities within these dynamic AD environments. Research into these microbial communities will ultimately result in greater yields of methane in AD.

A greater understanding of the microbial communities can be achieved via metagenomics, which involves the study of genomes recovered from environmental samples. Metagenomics involves the use of shotgun sequencing. Environmental DNA is sequenced followed by binning, and assembly into metagenome-assembled genomes (MAGs). Functional annotation is carried out to predict the gene function within the MAGs. However, quality and completeness of MAGs varies greatly due to the nature of shotgun sequencing. Large datasets of metagenomic data require large-scale data manipulation and bioinformatic analysis. Genome annotation pipelines (via workflow management tools e.g. Snakemake) allow automation and ensure reproducibility of the genome annotation.

A genome annotation pipeline was developed, using Snakmake, to predict the gene function of MAGs recovered from AD. This pipeline was developed to provide an automated tool to functionally annotate MAGs, in order to discover more about the metabolic processes and relationships between microbes that drive the AD process. A confidence system was devised to indicate the quality of annotations provided by orthology-based tools EggNOG and KofamScan, allowing further analysis of low quality ORFs. Reproducibility and reference databases continue to be limitations of bioinformatic pipelines. However, approximately 50% of ORFs are annotated to a high confidence.

Contents pages

List of Figures

List of Tables

Acknowledgements

I would like to thank my Supervisor, Professor James Chong for his consistent support and guidance throughout this research project and for offering me the opportunity to carry out this research. I would also like to thank Dr Sarah Forrester and Annabel Cansdale for passing on their knowledge and expertise, along with their time, to help me develop the skills necessary to complete this research project. Finally, I would like to thank Dr Emma Rand, as my thesis advisory panel member, for offering critical advice and feedback that has helped to improve the development of this research project.

Declaration

I declare that this thesis is a presentation of original work, and I am the sole author and that it was conducted by myself under the supervision of Professor James Chong. This work has not previously been presented for an award at this, or any other, University. All sources are acknowledged as References. The metagenome-assembled genome (MAG) data was processed by members of the Chong lab.

Chapter 1: Introduction

1.1 Anaerobic Digestion and its Environmental Impacts

Anaerobic digestion (AD) is an integral part of the wastewater treatment process in the UK that generates electricity from biogas production. Primary solids and activated sludge are fed into an anaerobic digester under anaerobic conditions. Microorganisms in the digester metabolise anaerobically, producing biogas composed mainly of methane, carbon dioxide. Biofertilisers are also produced from the remaining biosolids. The production of biogas was estimated as over 1 million m^3 biogas in 2019, where 24% of biogas was from farm sites and 28% on commercial sites was integrated into the national grid (WRAP, 2020). The methane produced during AD is then injected directly into the National Grid and can be converted into electricity for commercial and domestic use. The production of biofertilisers that are derived from organic waste have the potential to replace artificial fertilisers which are known to have a wider impact on ecosystems, including causing eutrophication (Yang, *et al.*, 2008).

The UK government has committed to ensuring that the UK does not contribute to increasing global temperatures beyond 2°C by signing the Paris agreement in 2015 (United Nations / Framework Convention on Climate Change, 2015). There is a general decreasing trend in Carbon Emissions (Figure 1) in the UK (DUKES, 2021), particularly in the last 10 years. The greatest difference is in the usage of coal over this time. There has been a decrease in petroleum usage since 1990, although this is not a massive change. Petroleum is still used to run cars. The overall decrease in emissions in 2020 would have been due to the global pandemic, which saw commercial airlines grind to a halt, decreases in commuting via personal vehicles, and less need for public transport. However, by 2021 that use in petroleum is back up to what is typically expected (Figure 1). The rise of electric cars should start to drive a reduction in petroleum consumption. However, there is a need to make electric cars more affordable for the average household, as currently the upfront cost for an electric car steers buyers away from the more environmentally friendly option. This trend in the data looks to comply with the promise that the UK government made to minimise its contribution to global temperature rise, aligning with the Paris Agreement, which aims to limit this rise to 2°C by 2050.

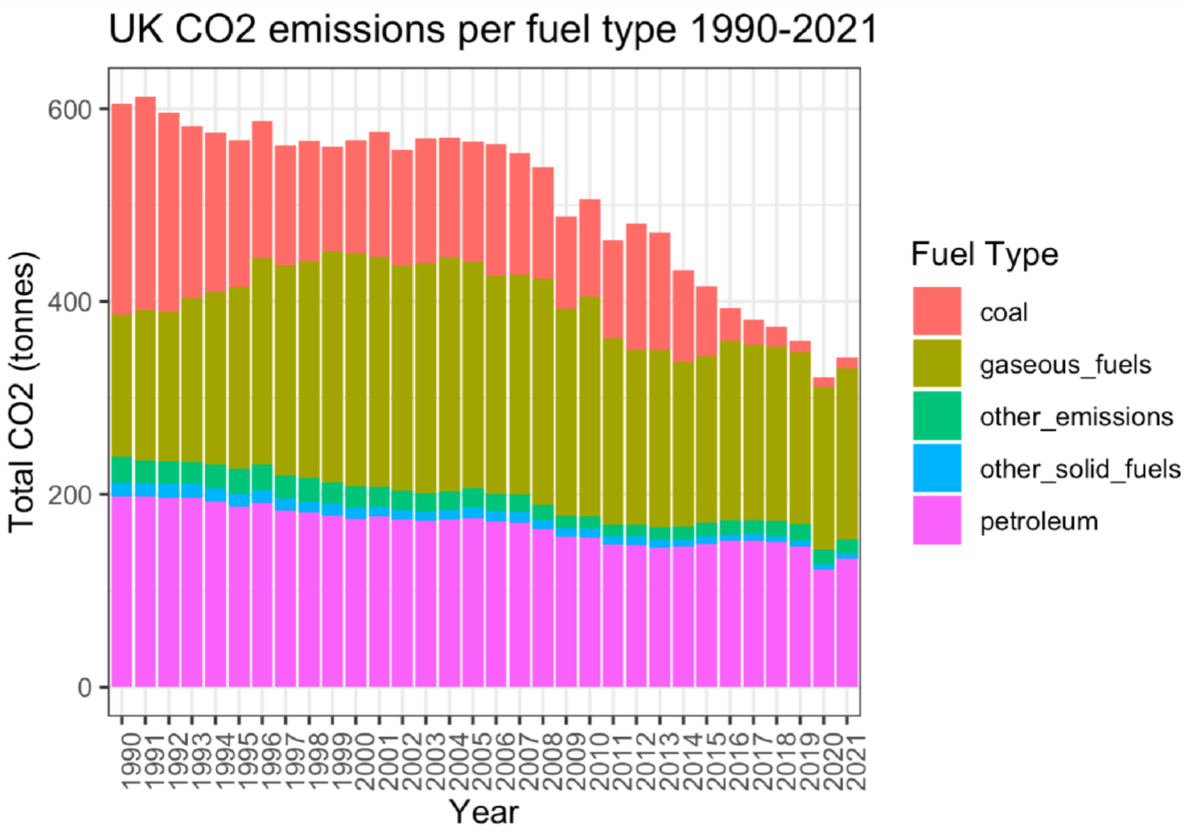


Figure 1. bar plot illustrating the UK Carbon Emissions (by fuel type) from 1990-2021. The data was provided by a publicly available report by the Digest of UK Energy Statistics (DUKES, 2021) and the plots were generated in R Studio using ggplot2.

The Global carbon emissions of the energy utilities company Shell Energy (Shell Energy, 2021) illustrate that there are efforts to reduce emissions. The amount of methane (CH_4) emitted (globally), by the company, has decreased between 2017 and 2021 (Figure 2). The total amount of carbon dioxide (CO_2) that was emitted by Shell from 2017 to 2021 (globally), similar to that of methane gas, was reduced (Figure 3). However, the data sets were not presented within the same units as the total CH_4 emissions is 10^3 more than CO_2. This does not make the results incomparable, as it shows that the overall difference in CH_4 emissions is much less significant when compared to that of CO_2 emissions. These data sets are worth further comparison as the Global Warming Potential (GWP) of CH_4 is 81.2 times higher than CO_2 over a 20-year period (Smith, *et al.,* 2021). Although there are some efforts to lower carbon emissions, as seen above in the UK, but also across international companies, these efforts still need to cut carbon emissions significantly. This is to ensure global temperatures do not reach the 2°C increase, causing significant and detrimental effects on the planet from extreme weather events i.e. droughts that would then negatively impact an already struggling food supply and therefore demand would exceed what is available.

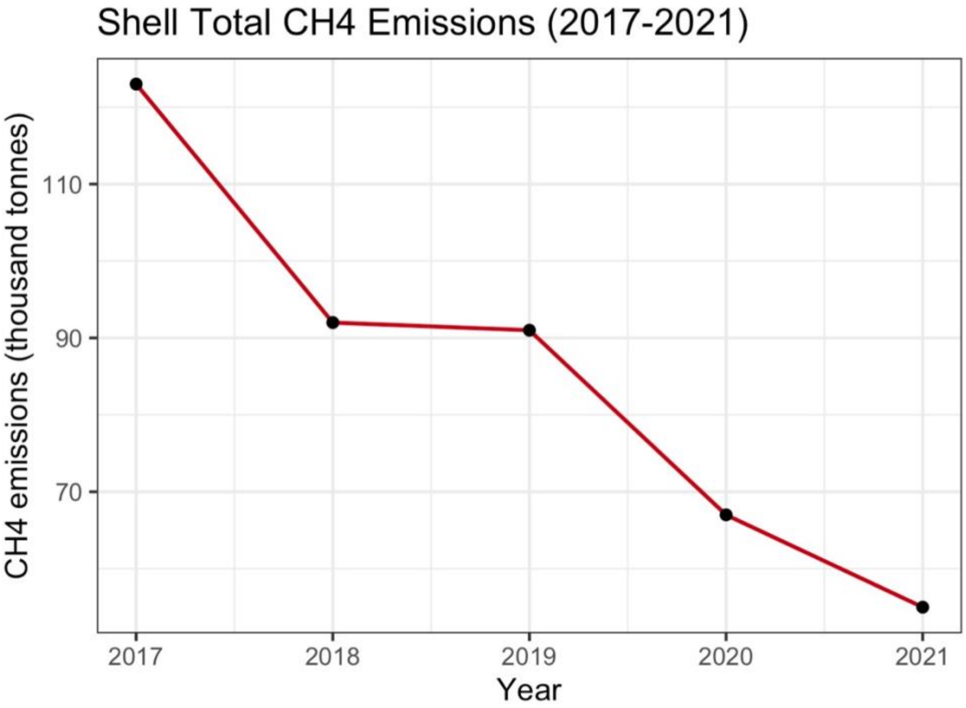


Figure 2. Plot illustrating the methane (CH_4) emissions (thousand tonnes) generated by Shell Energy from 2017-2021. Data is publicly available (Shell Energy, 2021), and the plots were generated in R Studio using ggplot2.

The total change in CO_2 emissions (Figure 4) that were emitted by BP energy company (BP, 2021) from 2011-2021 (global) showed that there was no overall trend of this data. However, there was a significant drop in CO_2 emissions from BP in 2020 this is indicative of the COVID-19 pandemic and therefore the lower demand for carbon as an energy source.

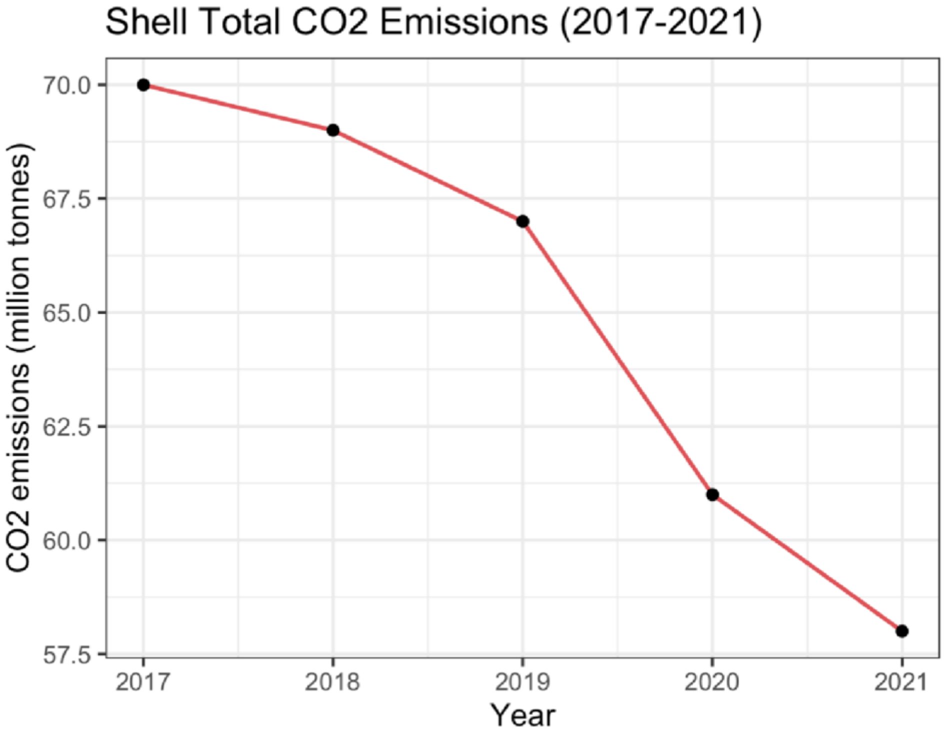


Figure 3. Plot illustrating the total carbon dioxide (CO_2) emissions (million tonnes) generated by Shell Energy from 2017-2021. Data is publicly available (Shell Energy, 2021), and the plots were generated in R Studio using ggplot2.

Alongside this Paris Agreement, there are other environmental targets such as the 17 Sustainable Development Goals (SDGs) developed by the United Nations, to be achieved by 2030 (UN General Assembly, 2015). These goals were established to tackle some of the World's greatest challenges from poverty to climate change. SDG number 7, for example, aims for "Affordable and Clean Energy" by 2030, in order to achieve this goal the productivity and affordability of renewable energy sources, such as AD, needs to be improved. Improving the efficacy of renewable energy sources will enhance global efforts to meet the Paris Agreement.

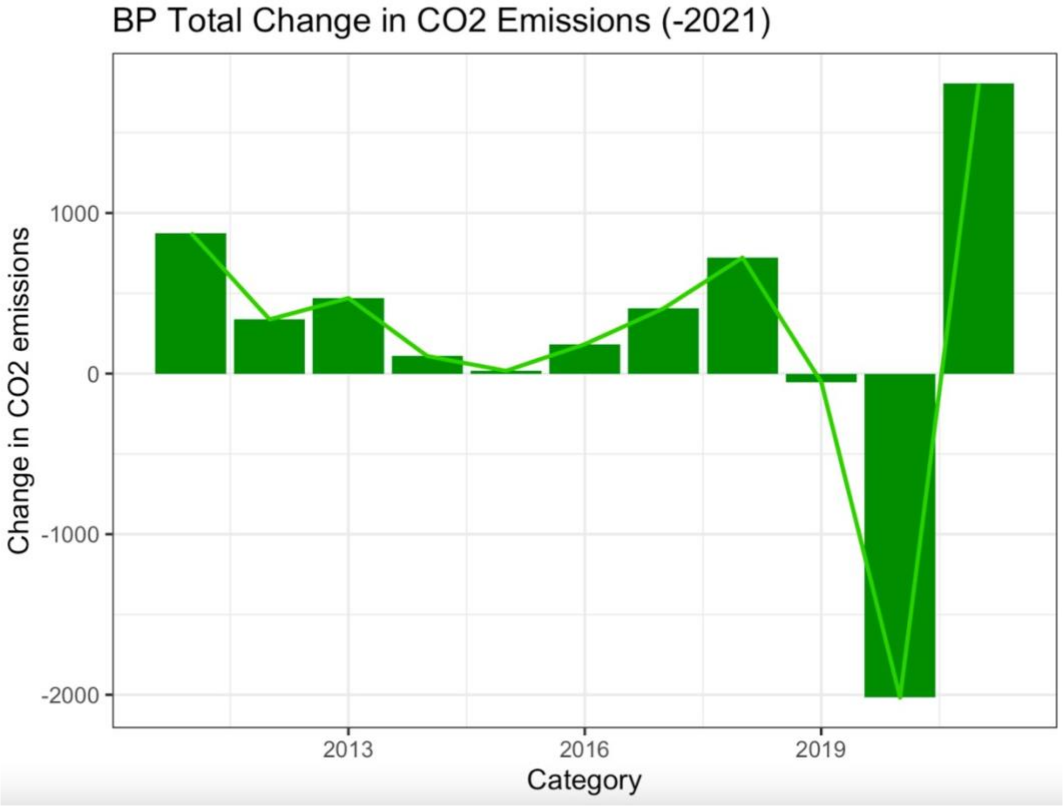


Figure 4. bar plot illustrating the total change in carbon dioxide (CO_2) emissions generated by BP from 2011-2021. Data is publicly available (BP, 2021), and the plots were generated in R Studio using ggplot2.

Additional benefits of increasing renewable energy usage include social advantages, such as providing many opportunities for environmental job opportunities. This would in turn improve skills and education to communities and therefore, improve income and increase investment in greener infrastructure subsequently building up local economies. This aligns with SDG number 9, which aims to "build resilient infrastructure, promote inclusive and sustainable industrialisation and foster innovation". SDG 11 is also of note here, which aims to "Make cities and human settlements inclusive, safe, resilient and sustainable". AD involves converting the solids of wastewater into biomethane as a source of renewable energy, improving the sustainability of UK cities, as well as reducing the disposal of harmful waste into the environment. This would lead to a safer environment for humans, wildlife, and local ecosystems.

Efforts to dispose of waste appropriately have been ongoing for a number of decades, from recycling bins for domestic waste, to disposing of industrial/commercial waste at recycling centres. The waste disposed of by Shell energy from 2017-2021 globally (Figure 5), does not show any significant change, although the data shows results indicative of the COVID-19 pandemic. The significant increase in disposal of hazardous waste is likely due to increased use of medical equipment, such as surgical masks (You, Sonne and Ok, 2020). Surgical masks were being used, not only in a medical setting but in everyday life, something that was not typically seen before 2020. This highlights the importance of appropriately and safely disposing of waste in a sustainable way, e.g. more of global organic waste could be incorporated into anaerobic digestion systems to be converted into biomethane.

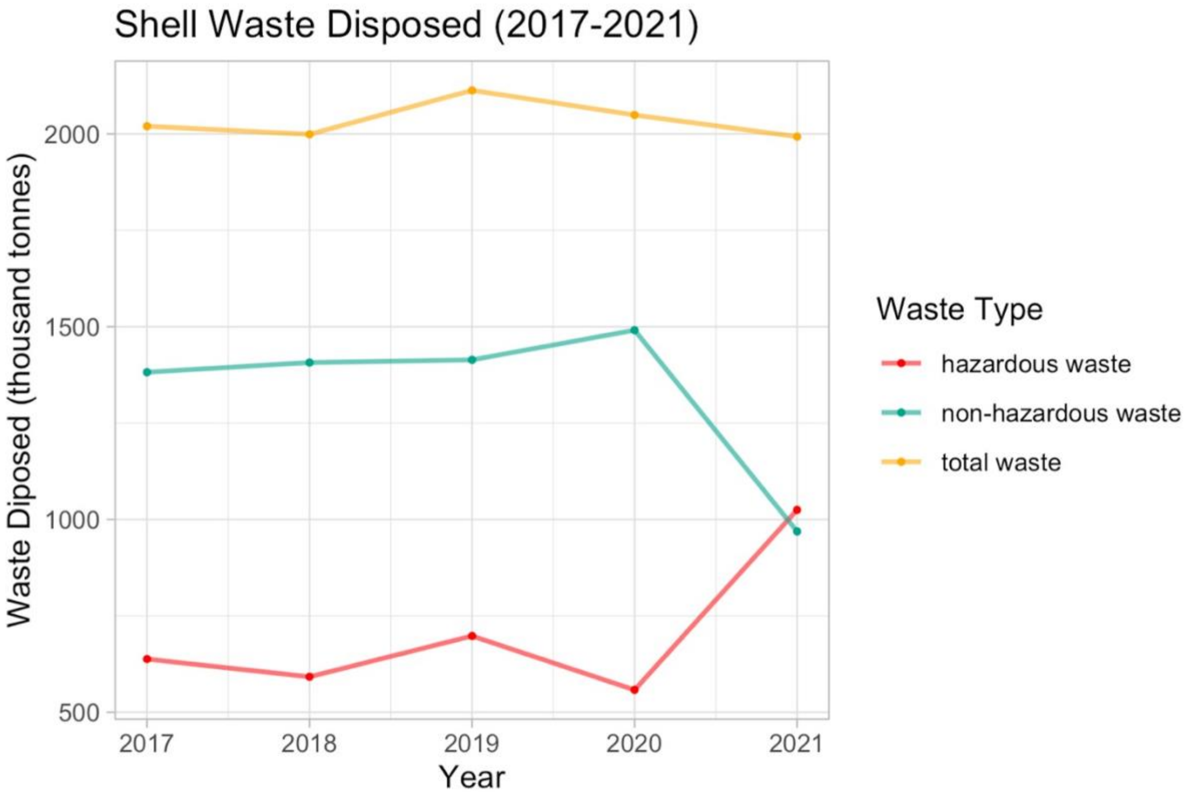


Figure 5. Plot illustrating the total amount of waste (hazardous and non-hazardous) disposed by Shell Energy from 2017-2021. Data is publicly available (Shell Energy, 2021), and the plots were generated in R Studio using ggplot2.

Although AD holds great potential as a renewable energy source it is also met with a number of limitations. Some of these limitations include the extreme unreliability of the AD process and small exposures to oxygen can cause devastating consequences on the system. Foaming also impacts the efficacy of the process and requires the AD system to be cleared out entirely. The clearing process is incredibly costly and time consuming (Jiang, *et al.,* 2018). Foaming can be caused by the rate at which organic matter is fed into the digesters or the presence of filamentous bacteria i.e. *Microthrix parvicella (*Westlund, Hagland, and Rothman, 1998*)*. These bacteria are found in higher abundance when the metabolites that cause foaming are also in high abundance. The demand for energy across the globe is not yet matched by the current renewable energy sources (such as, AD) but is currently met by non-renewable energy. Until the demand for energy can be met, the use of non-renewable energy sources (e.g. gas and coal) will continue to be the primary source of energy. It also does not come close to the financial gain of fossil fuels such as coal and gas despite them causing the most detrimental effects on the environment (*S&P Global Market Intelligence,* n.d.). Therefore, it is vital that bioinformatic pipelines of microbes in AD are developed to decrease the time spent researching and ultimately improve AD's capabilities as a renewable energy source.

1.2 The Anaerobic Digestion Process

AD depends heavily on the dynamic and complex microbial community within the digesters, often determining the productivity of biogas production (Nguyen, Nguyen and Nghiem, 2019). AD consists of four consecutive steps: hydrolysis, acidogenesis, acetogenesis and finally methanogenesis (Meegoda, *et al.,* 2018). Hydrolysis involves the conversion of products such as carbohydrates, proteins and lipids into monosaccharides, amino acids and fatty acids respectively, via hydrolytic bacteria (Bajpai, 2017). Next in the process is acidogenesis, which is the production of intermediates, including Volatile Fatty Acids (VFAs), via acidogenic microorganisms, from the up-take of the constituent products produced in hydrolysis (Detman, *et al.,* 2021). Then acetate is typically generated by the reduction and conversion of larger, more complex VFA molecules into acetate, via acetogens. Finally, acetate, as well as CO_2 and H_2, are metabolised by methanogenic archaea resulting in the formation of methane (Mahmood, Q. *et al.*, 2006).

The ratio of hydrolytic to methanogenic microorganisms determines the kinetics of the AD process (Ma, *et al.,* 2013), therefore hydrolysis and/ or methanogenesis can be defined as the rate-limiting step. This highlights the importance of the composition of microbial communities in AD, which consequently highlights the value of gaining more insight into microbial communities, to ultimately improve AD's productivity as a renewable energy source. The findings by Ma, *et al.,* 2013 were initially determined using microbiological techniques by culturing microbes and then determining the rate-limiting step. This study involved dairy manure that was collected and passed through a sieve, to remove large solids, before being added into anaerobic digesters. Samples were separated into two groups, heated sludge and normal (untreated) sludge. Here, the heated sludge was heated to 105°C for 2 hours to destroy non-spore-forming methanogenic archaea in the sludge. A modified Biochemical Methane Potential (BMP) assay was developed to determine the ratelimiting step of AD. The BMP assay consists of a mixture of heated sludge, glucose and sodium acetate. A respirometer was then used to record the production of methane gas. The methane production of the normal sludge (untreated) sludge was used as a control.

1.3 Microbial communities within biological systems

Common microbes found in AD microbial communities include a number of acidogens such as firmicutes (e.g. *Clostridium difficile* and *Bacillus subtilis*), bacteroides (e.g. *Bacteroides fragilis*), and proteobacteria (e.g. *Pseudomonas aerunginosa*) (Nguyen, Nguyen and Nghiem, 2019). These bacteria are all anaerobic. Some are obligate anaerobes, such as *Bacteroides fragilis* and *Clostridium difficile*, and others are facultative, like *Bacillus subtilis*. Many acetogens (microbes that produce acetate via anaerobic processes) including bacteria from the Syntrophomonas genus can also be found in AD systems. Finally, a major contributor to the microbial community are methanogens that can produce methane gas under anaerobic conditions. These archaea are anaerobic microbes that are typically found in a number of extreme environments (extremophiles).

Methanogens are methane-producing archaea that broadly share homology, as they typically belong to the same phylum, *Euryarchaeota*. However, recent research into the methanogenesis pathways indicate that the pathway could be present in two other phyla *Bathyarchaeota* and *Verstraetearcheota (*Lyu and Liu, 2019*)*. Within the *Euryarchaeota* phylum there are seven different orders, *Methanosarcinales*, *Methanobacteriales, Methanomicrobiales, Methanococcales*, *Methanopyrales*, *Methanocellales and Methanomassiliicoccales (belong to the phylum Thermoplasmata)* that methanogens are currently classified by. These taxons were determined by comparing 16S rRNA sequences and physiological features, e.g. morphology, cell structure, of methanogens (Liu, 2010). The wide diversity of methanogens highlights the importance of characterising methanogenesis in a variety of methanogens. This would ultimately improve the efficacy of the fourth and final step of AD, methanogenesis, as the methanogenesis process would be well researched and its intricacies would be more understood. AD digesters could then be manipulated to favour certain metabolic pathways. This could then produce a higher yield of biogas at a more affordable cost. An example that shows the importance of understanding homology in methanogens, is the taxonomy of *Methanothrix soehngenii*, which was unclear for many years. However, recent analysis of its physiology and morphology, via electron micrographs, in comparison with other methanogens finally defined its taxon (Jetten, Stams and Zehnder, 1992) and was re-identified as the *Methanosaeta* genus. *Methanosaeta* and *Methanosarcina* are common methanogens found within AD. *M. soehngenii* is more prevalent in the process of anaerobic digestion as it has a high affinity for acetate but has a higher yield in lower acetate concentrations. With reference to enzymatic activity, *M. soehngenii* uses acetyl CoA synthetase to activate acetate in the initial step of acetoclastic methanogenesis whereas as *Methanosarcina spp.* uses acetate kinase, which has a greater activity in higher acetate concentration. Thauer et al characterises the methanogenesis pathways used by different methanogens such as *Methanosarcinales* and *Methanococcales* (Thauer, 1998).

There is a syntrophic relationship between acetogens and methanogens in AD, where the acetogens produce acetate and hydrogen that allow the methanogens to metabolise these products into methane (Harirchi, *et al.,* 2022). There are three common types of methanogens; acetoclastic, methylotrophic and hydrogenotrophic. Each of these produce methane via different metabolic pathways. During acetogenesis, there is often an increase in hydrogen leading to a lower pH in the digester (Wang, *et al.,* 2021). The excess hydrogen can be used up during hydrogenotrophic methanogenesis via hydrogenotrophic methanogens, or during the production of butyrate and propionate (which are common VFAs produced in AD) (Darwin and Cord-Ruwisch, 2019). However, excess VFAs can disrupt the AD digesters causing failure (Yu and Fang, 2003). Also, acetogenesis is generally thermodynamically unfavourable (Harirchi, *et al.,* 2022) however, this process is made possible via sulphate-reducing bacteria (SRB). This ensures hydrogen levels are kept at an optimum (de Lemos Chernicharo, 2007). Hydrogenotrophic methanogens are common SRBs in AD, this transfer of hydrogen between SRBs and acetogens is a crucial interaction that allows AD to occur. The understanding of these complex interactions occurring in AD is vital to improve the productivity of methane production.

The acetoclastic methanogenesis pathway is the primary metabolism of *M. soehngenii* and is well characterised. This pathway consists of a multi-step process involving the conversion of acetate into methane, which is catalysed by multiple proteins (enzymes) produced by genes found in the genome of *M. soehngenii*. Genes of interest in this pathway include mcrA gene, which is typically used to identify methanogens (Friedrich, 2005). McrA (methyl coenzyme M reductase A) catalyses the conversion of methyl-coenzyme M into methane. The abundance of the mrcA gene shown in Morris, R. *et al* shows a *corresponding* increase in the production of methane by this organism under these conditions. This increase in methane production can be understood with reference to Ma, J. *et al*, as the methanogenic microbes are shown as rate determining factors in the AD process (Morris, *et al.,* 2014, Ma, *et al.,* 2013).

Efforts to gain a greater understanding of microbial communities in AD systems are ongoing. However, there are microbial communities that are more highly researched such as the gut microbiota. The increasing interest in the gut microbiome is due to the profound impact that these microbes have on the human body from digestion, to the immune system, to mental health conditions. The gut microbiota consists of commensal bacteria that in typical proportions, aid certain functions of the human body. *Clostridium difficile* is an example of a commensal bacterium, which is typically harmless in the correct proportions. However, administration of high levels of antibiotics affects the proportions of the gut microbiota, allowing *C. difficile* to thrive. This causes a *Clostridium difficile* infection, antibiotic-associated diarrhoeal disease (Schäffler and Breitrück, 2018), highlighting the delicate nature of the microbial communities. *C.difficile* allows the digestive system to thrive when present in low abundance but when present in high abundance, the bacterium harms the entire system. Microbes are also capable of having such a profound effect on AD systems, as the composition of microbes drives the AD process.

A synthetic gut model (Vrancken *et al.*, 2019) was designed to imitate the environment inside of the human gut. This will allow for rapid progress in the understanding of this greatly complex microbial community. A similar concept to the synthetic gut model is System 60, found at the University of York, which is a controlled model for AD systems. This is beneficial to the study of AD, as AD is difficult to control at an industrial scale. In addition, the use of System 60 can be more beneficial than long term studies of the ever-changing human gut microbiota, as these studies are easily affected by varied food intake, antibiotic intake, and several other factors. The experimental data obtained from AD models like System 60, alongside the findings from the metagenomic data will ultimately further the improvement of AD productivity. Easy Manipulation of the conditions of System-60 digesters, e.g. temperature, pH, pressure, or the addition of gases or feed at certain time points; will help to determine the optimum conditions for AD digesters. Sequencing samples taken from the digesters at certain time points, will help to discover the composition of microbes that thrive at these particular time points, which will also lead to improving AD, as favouring certain metabolic pathways can ensure a desired end-product.

1.4 The applications of metagenomics

In the past, inoculation and cultivation techniques were used to identify microbes in AD. Then, more advanced molecular biology techniques were used to understand the microbial communities in AD. These included DGGE, PCR and cloning on marker genes, molecular fingerprinting, qPCR, etc. (Lim, *et al.*, 2020). Although these molecular biology techniques paved the way to understanding these communities, they provided relatively basic findings in comparison to the information that has been discovered within more recent years. This significant increase in research and therefore understanding of the AD process, has been made possible by the development of multi-omic techniques and analysis, including metagenomic analysis (Zhang, *et al.,* 2017**,** Zhang, *et al.,* 2019). Metagenomics is the analysis of genomic data that was recovered from environmental samples, and provides predictions of all of the genes found within a genome or genomes.

Efforts to culture microbes from extreme environments is generally very challenging; however, the application of metagenomics provides genomic data without the need to culture individual microbes in the laboratory. This is especially advantageous for microbes that are unculturable. The capacity for new information about certain species and evolutionary genetics etc. has been expanded massively with the application of metagenomics. This allows for the study of complex dynamic communities within anaerobic digesters, soil, oceans, and glaciers; and is also being applied within the medical field (Pérez-Cobas, Gomez-Valero and Buchrieser, 2020). The application of metagenomics has also greatly benefitted the medical community, by discovering more information of pathogenic microbes within these communities. This will help to develop new medical treatments (Zhang, *et al.,* 2021).

Metagenome-assembled genomes (MAGs) are genomes recovered from environmental samples, which involves the assembly and binning of metagenomes. Shotgun sequencing is a technique often used in assembly of MAGs (Gerlach and Stoye, 2011) where an environmental sample, containing microbial genomes, is collected. The DNA is then extracted and fragmented (Ghurye, Cepeda-Espinoza and Pop, 2016) so that a library of fragments can be subjected to DNA sequencing. Genomes are sequenced as either long reads or short reads, and this is typically decided by the technologies that are available at the time, and often the cost also (Olson, *et al.,* 2019). The random collection of reads are then assembled, via alignment by comparison against reference genomes, into contigs (Nissen, *et al.,* 2021). The contigs are binned by grouping contigs of the same taxon, by using taxon-dependent methods, that use reference databases, or computer models of predicted sequences, to compare the contigs against. Contigs that are not comparable to either the databases or computer models are binned as unassigned (Strous, *et al.,* 2012). This allows for taxonomic classification of contigs and is often used for further functional analysis as well. Challenges associated with binning and alignment include a bias of taxonomic groups available in reference databases during these processes. Well-researched taxons are typically prioritised over less-understood taxons, impacting accuracy of binning to the correct taxonomic level (Wickramarachchi and Lin, 2022).

MAG sequences can be annotated revealing the function and taxonomy of the genes found in the MAGs. This is typically carried out by comparing MAGs to databases (such as Kyoto Encyclopaedia of Genes and Genomes- KEGG). Common functional annotation (bioinformatic) tools include EggNOG (Huerta-Cepas, *et al.,* 2019), GhostKOALA (Kanehisa, Sato and Morishima, 2016), InterProScan (Jones, *et al.* 2014), BLAST+ (Camacho, *et al.,* 2009). These tools can all be applied to metagenomic data in order to predict protein function. EggNOG and GhostKOALA use the KEGG database and can then compare genes found in metabolic pathways to KEGGmapper (Kanehisa and Sato, 2020). Methanogenesis pathways have been recognised and inputted into the KEGG database, making the identification of methanogens and subsequent genes involved in methanogenesis relatively straightforward. Although some of the genes involved in methanogenesis are present in some bacteria, identification of the entire pathway, or partial identification of the pathway, should be considered when identifying microbes.

The extraction of DNA from environmental samples and shotgun sequencing removes the need to culture microorganisms individually (as this is typically not possible), and allows the extraction of DNA from dead cells (Gerlach and Stoye, 2011). Shotgun sequencing followed by the assembly into genomes, produces Metagenome-Assembled Genomes (MAGs). Completeness of MAGs varies greatly (Nayfach, *et al.,* 2021), typically because of the nature of shotgun sequencing and further binning into contigs, making functional annotation tools less accurate, as it is likely that significant parts of the MAG sequences are misplaced. It is also likely that some microbes might be missed from the samples being sequenced as they are only found in low quantities (Gasc and Peyret, 2018) within these microbial communities. This can provide an inaccurate consensus of the entire microbial community as not all of the microbes present will be accounted for within the samples. An inaccurate consensus of the entire AD community can also be caused by cells that are potentially very fragile (the DNA can be damaged) or cells that are too tough to penetrate (no DNA can be recovered). Although the data generated from these datasets is not indicative of the entire environment, they provide sufficient information about the microbes found in the environment, and the processes they are responsible for. Contamination is also a massive challenge to overcome when carrying out shotgun sequencing (Chen, *et al.,* 2020). This occurs when fractions of DNA from several organisms within the samples collected are mistakenly involved in the assembly of a particular MAG. There are several common errors in genome assembly. One of these errors is the incorrect assignment of repeated DNA segments within the same MAG or other closely related MAGs in the environmental sample (Olson, *et al.,* 2019). The development of bioinformatic tools, such as FastQC (Andrews, 2010), can be used to highlight contaminants within metagenomic sequences, leading to their removal. The outputs from current bioinformatic software are sub optimal (Salzberg, 2019), as the overall standard of MAGs is currently inadequate and becomes a limiting factor. However, this can be assessed through quality metrics via software such as CheckM (Parks, *et al.,* 2015).

A study collected 1635 MAGs from 134 datasets of MAGs, from various different anaerobic digestion reactors (Campanaro, *et al.,* 2019). Functional annotation via metabolic reconstruction led to the discovery of the role of the microbes from AD in a number of processes. These processes include methane production and the conversion of organic waste into productive biomass (e.g. biogas and biofuel). This study highlights the importance of metagenomic data in achieving a greater understanding of the complex processes that make up the AD process.

The use of metagenomics on AD data has already significantly improved the understanding of the microbial communities within AD systems, by identifying a greater number of microbial species present and therefore their function within the complex community (Maus et al., 2017). This understanding will lead to greater yields of biogas production in AD.

1.5 The use of workflows and pipelines for metagenomic data

Analysing large datasets involves generating useful information from raw data. In the context of bioinformatics this means analysing the data through multiple bioinformatic tools and reference databases. Many challenges are faced when producing scalable and reproducible results (Perkel, 2019). These results are easily transferred onwards into further analysis, such as converting data into a versatile format (FASTA), and improving previous analysis from bioinformatic tools such as comparative genomic and taxonomical analysis. As a result, bioinformatic data and analyses now rely heavily on the use of workflow management tools. Workflow tools allow the automation of pipelines and decrease the time taken to develop these pipelines (David and Glore, 2010). Workflow management tools also allow the execution of multiple commands and tools in order to carry out an overall analysis (Wratten, Wilm, Göke, 2021). In this instance the analysis is carried out on genomes (or metagenomes). The automation of these tools offers improved scalability and productivity (Perkel, 2019) in the research space as additional work can be achieved while the workflow tool is running the desired analyses.

A big concern when performing large-scale bioinformatic analysis is reproducibility (Perkel, 2019). Problems typically occur due to minor changes in individual operating systems or computers. Another common issue is the lack of organisation (Wratten, Wilm, Göke, 2021) with reference to the bioinformatic tools used and their respective dependencies, which often cause clashes (David and Glore, 2010). Workflow management tools such as Snakemake (Köster and Rahmann, 2012), Nextflow (David and Glore, 2010) and Galaxy (Jalili, *et al.,* 2020) allow for container-like environments. This is similar to the environment a high performance computing cluster offers when Conda environments or Docker are utilised. The containerisation allows for these tools to integrate external bioinformatic tools into the workflow, without the dependencies of certain tools breaking the pipeline by prohibiting some parts of the pipeline to work. Multiple Conda environments can be implemented into a pipeline when using workflow management tools, typically each tool will have its own Conda environment, to ensure the resolution of this issue.

Snakemake (Köster and Rahmann, 2012) is a portable workflow management tool that can be utilised on a local device as well as a Cluster (like Viking). Snakemake is a modular tool based on rules. All of the rules collectively achieve an overall goal, which is set by the first rule, referred to as 'Rule all'. This Python-based tool is accessible via command line, and clear and concise messages are provided to the user to indicate errors within the pipeline (which can quickly become excessively complicated).

High Performance Computing (HPC) clusters can play a crucial role in reproducibility of multi-omic analyses, as the environment is the same for all users. This benefits users as it eliminates the need for high-performance personal computers, which are often expensive and not essential for the typical biologist.

This does eventually become a limitation of the pipeline, in terms of reproducibility, as only users that have access to the HPC cluster would be able to utilise the software or datasets that are developed or analysed respectively. Other than HPC users, those with a similar computer set-up to the HPC would also be able to utilise the software and datasets developed, highlighting the importance of noting operating system versions and software versions. Finally, some bioinformatic tools require an increased memory-usage or more CPUs, which may exceed the limits of a personal computer, reducing the speed of the tool or rendering it unusable. HPC clusters allow the allocation of different amounts of memory and CPU cores to run the tools.

There are many genome annotation pipelines already publicly available, such as PGAP (Tatusova, *et al.,* 2016), MetaErg (Dong and Strous, 2019), Microbe Annotator (Ruiz-Perez, Conrad, and Konstantinidis, 2021), and FA-nf (Vlasova, *et al.,* 2021). PGAP is a prokaryotic genome annotation pipeline developed by NCBI and was automated using Common Workflow Language (CWL). It is common for genome annotation pipelines, including the pipelines mentioned above (Table 1), can be installed locally and onto HPC clusters. The typical data files that are required for these pipelines are FASTA sequences. All of these pipelines carry out homology-based predictions, however, PGAP does not carry out orthology-based predictions, like MetaErg and FA-nf.

Table 1. A Comparison of features in current functional annotation pipelines. The pipelines include PGAP, FA-nf, MetaErg and Microbe Annotator.

Categories	**Features**	**MetaErg**	**PGAP**	**Microbe Annotator**	**FA-nf**
Use	Local installation	Yes	Yes	Yes	Yes
	Command line/ Cluster	Yes	Yes	Yes	Yes
Software	Written using:	Docker	CWL	Python-based	Nextflow
Databases	Multiple used	Yes	Yes	Yes	Yes
	KEGG Orthology	Yes	-	Yes	Yes
	RefSeq	Yes	-	Yes	-
	TIGRFAM	Yes	Yes	-	-
	Pfam	Yes	Yes	-	Yes
	SwissProt	Yes	-	Yes	-
	BLAST	-	?	-	Yes
	InterPro	-	-	-	Yes
Functions	Homology-based predictions	Yes	Yes	Yes	Yes
	Name-based predictions	-	Yes	-	-
	Orthology-based predictions	Yes	-	Yes	Yes
	Pfam assignments	Yes	Yes	Yes	Yes
	KEGG KO/pathway assignment	Yes	-	Yes	Yes
Outputs	HTML results page (interactive plots)	Yes	-	-	-
	Summary report for Genbank submission	-	Yes	-	-
	Annotation tables of modules completion	-	-	Yes	-
	Annotation tables from each tool	Yes	Yes	Yes	Yes

Although these pipelines all result in the same overall outcome, the process of retrieving the information differs and the format of results differs. The use of multiple different databases ensures that the maximum number of proteins are annotated per query sequence. For example, Microbe Annotator will compare query sequences to each database until a KEGG KO number is assigned. However, sequences not assigned a KEGG KO are discarded and only 48% of proteins within the KEGG database have been assigned a KEGG KO (Aramaki, *et al.*, 2019). Workflow management tools are typically used to develop pipelines e.g. FA-nf used Nextflow and PGAP used CWL; however, none of these tools use Snakemake. Snakemake is the Workflow management tool that is intended to be used for the pipeline developed during this research project. To conclude, the requirements for these functional annotation pipelines are simple and minimal, which provides an easy process for the efficient conversion of metagenomic data into useful information, via a thorough analysis.

As previously discussed, pipelines developed by workflow management tools improve reproducibility, scalability, efficiency; and are completely customisable. These workflow management tools allow for easier production of bioinformatic pipelines, hence there are already so many pipelines publicly available. However, production of a pipeline that is entirely specific will further reduce the risk of any challenges in relation to reproducibility appearing etc., as well as increase productivity tenfold in the research group. This will ultimately lead to a greater understanding of the role of microbial communities in anaerobic digestion.

1.6 Confidence of results and benchmarking

Benchmarking is a vital part of the development of bioinformatic tools as they help to provide insight into the accuracy of the tool by comparing the results against pre existing tools of a similar function (Urbanowicz, et al., 2018). Benchmarking can be used as a confidence measure of the results generated by bioinformatic tools. The Confidence measure can be quantified, this can be quantified by comparing total number of outputs generated, or, often, time taken to complete the analysis e.g. Kraken2 benchmarking against Kraken (Wood, Lu and Langmead, 2019).

Another form of quantifying the quality of the generated results is via a statistical method. For example, EggNOG-mapper (Huerta-Cepas, et al., 2017) provides expected-values, these E-values are a numerical value to compare the matches provided by the tool. The closer the E-value is to zero, the more likely the sequence made by the tool is expected to be present in the database. However, this method only compares the performance of this individual tool rather than being compared against other tools that provide a similar output.

A confidence measure of the quality of outputs generated by one tool compared to the quality of another tool would provide more of an accurate measure of tool performance. Providing quality control of the results generated by these tools will help to make more accurate protein predictions, by collecting all possible information for each query sequence. Furthermore, improving the quality of bioinformatic data generated will result in greater, more frequent advances within the scientific community.

Confidence ranking that already exists in bioinformatic tools.. Variant Ranker (Alexander, et al., 2017) is a tool that ranks variants of genes, to ensure the prioritisation of variants is provided to researchers to apply these within an experimental approach. Statistical confidence measures are commonly used to improve the quality of genomic mapping and assembly (Servin, de Givry and Faraut, 2010). The Annotation Confidence Score (Yang, Gilbert and Kim, 2010) was also developed as a confidence measure during genome annotation of gene or ORFs, to determine the quality of the predictions made by bioinformatic tools.

Devising a confidence measure, that determines the quality of the protein predictions from two homology-based tools (EggNOG and KofamScan), will begin to ensure the quality of the results generated by this genome annotation pipeline. The quality of the results will be determined by whether both tools provide the similar results, and the same amount of information. Through general use of these two tools respectively, these tools often generated differing results for the same query sequence. The confidence measure will be quantified with a value between 1-6, value of 1 is the highest quality and 6 is the lowest quality.

1.7 Project aims

Microbial communities are still largely not well understood. This is typically due to a lack of information about the microbes found within the communities, and is often due to an inability to culture these microbes in a laboratory environment. With the rising popularity of shotgun sequencing and metagenomic analysis, this information is slowly being uncovered. Although there are still limitations, the increasing number, and quality, of bioinformatic tools and reference databases means that this information continually improves. Consequently, the production and automation of a genome annotation pipeline, via workflow management tools, should allow the analysis of large datasets at a faster rate, ultimately leading to further research into microbial communities within anaerobic digestion.

The overall aims of the project are:

- To develop an automated genome annotation pipeline that will allow us to gain a greater understanding of the wider microbial communities in the digesters.
- To devise a confidence ranking system where each ORF will be given a confidence score based on the completeness of the annotation.

Chapter 2: Pipeline introduction

2.1 PROKKA

PROKKA (Seemann, 2014) is prokaryotic rapid annotation tool that annotates FASTA sequences and provides a brief annotation by providing a name, and sometimes a description, of the protein, any unknown proteins to the tool are labelled as 'hypothetical protein'. The genomic annotation is carried out by a number of bioinformatic tools that are independent of PROKKA. These tools include Prodigal (coding sequences) (Hyatt, *et al.,* 2010), RNAmmer (ribosomal RNA genes) (Lagesen, *et al.,* 2007), Aragorn (transfer RNA genes) (Laslett, and Canback, 2004), SignalP (signal leader peptides) (Teufel, *et al.,* 2022), and Infernal (non-coding RNA) (Nawrocki and Eddy, 2013). Prodigal identifies the coordinates of these features and then the other tools predict the coordinates' function. Typically protein function is predicted by comparing genomic sequences against large databases. This same method of comparison is used here except PROKKA has a hierarchical system where the sequences are first compared with a small reliable database, followed by a slightly larger one and so on. PROKKA generates 10 different formats of the annotation to allow for further analysis.

2.2 EggNOG-mapper

EggNOG-mapper (Cantalapiedra, *et al.,* 2021) is a functional annotation tool based on the KEGG Orthology (Kanehisa, *et al.,* 2016) and the KEGG database. This tool predicts functions of proteins by comparing against orthologs, instead of other homologs which is true of NCBI BLAST+ (Camacho, *et al.*, 2009) and InterProScan (Jones, *et al.*, 2014). Typically orthologs retain their function post-speciation events, whereas paralogs do retain their function post-speciation events. Therefore the functional annotation provided by EggNOG should be more accurate. Predictions of orthology are determined at different taxonomic levels. This can provide more scope to the genes within the genomes that are being annotated. If the prediction is at a lower taxonomic level (closer to species-level) the more targeted the function can be to that specific genome. ORFs are provided with a more extensive annotation than PROKKA, by giving a functional COG, KEGG number, taxonomic origin, and often a name and free description. The free text description is generated via a text mining machine learning pipeline (Huerta-Cepas, *et al.*, 2019). Functional COGs (often referred to as orthologous groups) are clusters of approximately three homologous sequences, that have divergently evolved from a single speciation event. Therefore based on this theory that orthologs are more likely to retain the function long even after speciation events occur. During the development of EggNOG-mapper the benchmarking approach was used to ensure the performance of the tool is adequate compared to similar existing tools. Gene Ontology (GO) predictions by EggNOG have been compared against NCBI BLAST+ and InterProScan. These also provide a homology-based prediction of function so therefore are appropriate for benchmarking (Huerta-Cepas, *et al.*, 2019).The developers of EggNOG have used Orthobench2 (Trachana, *et al.,* 2014) and Quest For Orthologs (QFO) (Altenhoff, *et al.,* 2016) to estimate whether consistent updates and the addition of genomes to the KEGG orthology database affects the quality of the predictions that are given by EggNOG. Finally, as all bioinformatic tools and software are often updated and improved, EggNOG is consistently evolving and improving and are currently focusing efforts to improve analysis of metagenomic sequences.

2.3 KofamScan

KofamScan (Aramaki, *et al.*, 2019) is a command line tool that, similarly to EggNOG, is a functional annotation tool that makes predictions on functions by comparing nucleotide sequences against the KEGG Orthology database. The tool provides an E-value, a KEGG number, and a description of the protein. The annotation from KofamScan lacks some information in comparison to the annotation from EggNOG. However, KofamScan has the option, by using a command line flag to generate an output file that can be directly inputted into KEGG-Decoder. Then, a heatmap is generated to visualise the KofamScan outputs. KofamScan is a homology-based search tool (similar to BLAST+ and GHOSTX) (Suzuki, *et al.*, 2014), which differs from the other bioinformatic tools that have been developed by Kanehisa et al e.g. GhostKOALA and blastKOALA (Kanehisa, Sato, and Morishima, 2016). The KEGG numbers predicted by KofamScan are also compared against a database of profile hidden Markov models (pHMM), unlike GhostKOALA and BlastKOALA.

The comparison of KofamScan and EggNOG acts as benchmarking for the pipeline, acting as a quality checking process. Both tools are homology-based search tools that are based on the KEGG orthology (KO) that provide similar outputs and similar predictions via slightly different methods. It is also interesting to see whether these tools will predict the same function for certain genes. A limitation of these tools basing their predictions of the KEGG Orthology, is that only 48% (12,934,535) of protein sequences are within the KEGG GENES database (which contains 27,173,868 protein sequences), have a KEGG number assigned to them (Aramaki, *et al.*, 2019). This results in half of the genes annotated to be left unaccounted for. Efforts to improve this percentage of assigned genes need to be pushed to make these tools more reliable for functional annotations.

2.4 KEGG-Decoder

KEGG-Decoder (Graham, Heidelberg and Tully, 2018) is designed to input the outputs from a number of KEGG-KOALA tools, i.e. GhostKOALA, BlastKOALA and KofamScan, in order to visualise the completeness of metabolic pathways found within the KEGG Orthology of multiple genomes at once. The outputs are visualised by a heatmap And the capability of comparing multiple genome sequences at once provides an instant comparison of their metabolism. This tool was developed by manually defining metabolic pathways and the enzymes involved within the KEGGmapper pathways, this file can be found on their Github (KOALA_definitions.txt) (*Github*, n.d.). The tool is consistently being updated as users can request pathways of interest to be integrated into the tool, as long as the genes present in these pathways are present in the KEGG database.

2.5 Kraken2

Kraken 2 (Wood, Lu and Langmead, 2019) is a high-speed metagenomic annotation tool that provides a taxonomical analysis of metagenomes using k-mers alignment. The database for this tool (Wood and Salzberg, 2014) was curated to consider exact k-mer alignment as well as the lowest common ancestor (LCA) taxa. The taxa being that of all genomes that have that specific k-mer. Using this method was comparable, in speed, to BLAST (Altschul, *et al.*,1990), this comparison is seen as a benchmarking example of the performance of Kraken 2. To improve the overall speed of Kraken 2 in comparison to the first version of the tool (Kraken1) the memory usage of the tool was reduced by 85%. This reduction allows for a greater number of reference genomes to be utilised within the tool without negatively impacting the accuracy of the tool. Its performance and accuracy was assessed by collating a group of 50 genomes, of which there were reference genomes for, as stated by Wood, Lu, and Langmead. Then a group of 50 genomes that were not found within the first group of 50 genomes was also collated. When compared with different taxonomic classifiers both groups showed consistent results. This tool is used in the manual annotation to provide some information to the ORFs with a lower confidence score in this pipeline. Simulated metagenomic data was generated in order to test the performance of the tool, by testing classification speed, as well as genus-level taxonomy accuracy. Simulated metagenomes were developed by combining two complete genomes from other projects. Illumina HiSeq and MiSeq sequencing platforms were used to sequence the data involved to produce these simulated metagenomes.

2.6 NCBI Blast+

NCBI Protein Blast+ (Camacho, *et al.*, 2009) is often the research standard for protein and microorganism identification, as it is routinely used as a benchmark for other homologybased tools such as KofamScan. Blast+ is a homology based search tool that compares query amino acid sequences to the BLAST database, by aligning query sequences to reference sequences from the database. An E-value (expected value) is then given to each matching sequence by calculating the number of matches that are expected to be found within the database (Camacho, *et al.*, 2009). An NCBI accession number is given to each matching sequence to provide a summarised report about each protein,, from its individual function, to the exact position in the plasmid/chromosome of the query sequences.

2.7 InterProScan

InterProScan (Jones, *et al.*, 2014) is a functional annotation tool that involves the analysis of DNA and protein sequences, and is available on command line. The homology of query sequences are compared against the InterPro database of reference sequences. The InterPro database is made up of multiple public databases and with every update more genome sequences are added. These databases often contain the same information within them such as functional sites, domain sites, and families of proteins (Jones, *et al.*, 2014). Multiple output formats are provided by the tool these include- xml, a text file and gff3, and this allows for further analysis of the sequence of interest.

2.8 SeqKit

SeqKit (Shen, *et al.*, 2016), the final tool in the pipeline, is a command line tool that manipulates nucleotide sequences in a FASTA or FASTQ format. The FASTA/Q format file is versatile and allows for sequence files to undergo further analysis without further manipulation of the file to be inputted into certain tools. The FASTA/Q format files are easily read with languages such as Python and Perl- languages that many common bioinformatic tools are written in. There are many FASTA/Q manipulation tools readily available, such as FASTX-ToolKit (Hannon, 2010), Pyfaidx (Shirley, *et al.*, 2015) and Seqtk (Li, 2012), that perform typical manipulations- filtering, converting, deduplication, searching, splitting and sampling. However, SeqKit offers more options than these tools e.g. shuffling, and locating motifs and common sequences. In total, SeqKit has 19 subcommands that are utilised on command line as flags which makes them easy to integrate into pipe commands to perform complex FASTA/Q manipulations. Therefore, one tool (SeqKit) can be used instead of multiple FASTA/Q manipulation tools.

Chapter 3 : Materials and methods

3.1 Data

The initial pass through of the pipeline was completed using a single known MAG, a putative *Methanothrix soehngenii* genome that was recovered from the Naburn wastewater site just outside of York. It was then binned and assembled into a MAG. This was chosen initially as the results from the bioinformatic tools were already known, and this will ensure that, not only the bioinformatic tools, but the pipeline is generating expected results. *M. soehngenii* is a methanogenic archaea, and is non-motile, non-spore forming, rod shaped, with an aggregate to form long filaments (Huser, Wuhrmann, and Zehnder, 1982), and is an acetoclastic methanogen that uses solely acetate as an energy source. *Methanothrix soehngenii* has been more recently renamed to *Methanosaeta concilii* upon revision (Patel and Sprott, 1990).

A second small dataset was generated by downloading a number of Genome sequences from NCBI nucleotide. This dataset is comprised of 5 highly characterised or researched prokaryotic organisms, these include *Escherichia coli* (NCBI Accession No. NC_002695.2), *Bacillus subtilis* (NCBI Accession No. NZ_CP103783.1), *Pseudomonas aeruginosa* (NCBI Accession No. NZ_CP102441.1), *Vibrio cholerae* (NCBI Accession No. NZ_CP080462.1), and finally *Staphylococcus aureus* (NCBI Accession No. NZ_CP064365.1). The idea of this small dataset is to pass the genomes through the pipeline and receive results that are expected, as these organisms are particularly well researched. These genome sequences are also high quality lab strains and thus there is a higher confidence in these sequences being complete genomes, as metagenomic data is more likely to be incomplete and not perfectly correct sequences. This is due to the reassembly of a larger mixture of DNA, that will ultimately result in DNA segments being misplaced. The results of this test dataset will also provide confidence in the annotation of the bioinformatic tools present in the pipeline and that the pipeline is outputting the results as expected.

All of the genomes in this dataset are prokaryotic bacteria, and are all facultative anaerobes, the majority (excluding *V. cholerae*) of which can be found in AD (Mukhuba, *et al.*, 2018). These bacteria typically undergo mixed acid fermentation under anaerobic conditions, among some other metabolic processes. Mixed acid fermentation involves the conversion of carbon based molecules into ethanol, acetate, succinate, formate, and lactate (Ciani, Comitini and Mannazzu, 2008).

3.1.1 Bacillus subtilis

B. subtilis uses sugars and other organic acids as an energy source. This occurs by either glycolysis, the pentose phosphate pathway, or the Krebs cycle (Schilling, *et al.*, 2007). However this is only in response to oxygen being present. As previously discussed, *B. subtilis* is a facultative anaerobe and is therefore able to undergo mixed acid fermentation under anaerobic conditions. With the use of high-performance liquid chromatography (HPLC), compounds such as lactate, acetate, and 2,3-butanediol, were identified as some of the main products of these fermentation processes (Cruz Ramos, *et al.*, 2000). Alternative electron acceptors (like nitrate, sulphate, and fumarate) can be used to undergo respiration under anaerobic conditions. It is hypothesised that nitrate ammonification is used by *B. subtilis* in low oxygen conditions where free electrons are donated to nitrate (Sun, Kokko and Vassilev, 2023).

3.1.2 Escherichia. coli

In the absence of oxygen *E. coli* undergoes fermentation producing products such as acetate, ethanol, lactate and formate (Förster and Gescher, 2014). Depending on availability of alternative electron acceptors, *E. coli* can undergo respiration under anaerobic conditions. With a lack of oxygen, however, the citric acid cycle is downregulated, resulting in an incomplete production of sugars, and acetate becoming the main product of this anoxic respiration (Förster and Gescher, 2014). Acetate is a fundamental compound of AD that drives acetoclastic methanogenesis, to produce methane gas (Pan, *et al.,* 2021).

3.1.3 Pseudomonas aeruginosa

P. aeruginosa undergoes arginine fermentation and pyruvate fermentation under anaerobic conditions. Pyruvate fermentation does not allow for the organism to thrive but allows for prolonged survival in the absence of oxygen (Kampers, *et al.,* 2021). Alongside anaerobic fermentation, *P. aeruginosa* undergoes nitrate dissimilation, and denitrification (where enzymes catalyse the transfer of electrons into nitrogen oxides). (Van Alst, *et al.,* 2007)**.**

3.1.4 Staphylococcus aureus

In conditions without oxygen *S. aureus* cannot use electron acceptors (oxygen) to undergo aerobic respiration, however alternative electron acceptors e.g. nitrate and nitrite, allow this process to occur anaerobically (Fuchs, *et al.,* 2007). Mixed acid fermentation is also undergone in anaerobic conditions to produce products such as lactate, acetate, formate, and 2,3-butanediol (Fuchs, *et al.,* 2007).

3.1.5 Vibrio cholerae

Under anaerobic condition *Vibrio cholerae* undergoes fermentation which is common in facultative anaerobes. However, in the presence of an alternative electron acceptor (AEA), *V. cholerae* can undergo respiration and fermentation simultaneously (Bueno, Pinedo and Cava, 2020). Common AEAs are nitrate and fumarate. Nitrate, nitrite, and sulphate, are AEAs commonly found within AD (Batstone, *et al.,* 2015). Therefore, respiration products of *V. cholerae* are likely present in anaerobic digesters.

3.1.6 NCBI Dataset Conclusion

The interest in the metabolic mechanisms of these organisms is to understand the breakdown of compounds in AD, to then produce high value products, such as methane gas. In addition, the metabolic characteristics of these organisms are visualised in the pipeline precisely, to ensure the pipeline is producing high-quality results.

3.2 File structure

An integral part of ensuring that a pipeline is working correctly and is therefore reproducible, is its filing structure. An emphasis was placed on ensuring the filing structure for this pipeline was concise and easy to navigate (Figure 6), especially considering that there is no GUI on the Viking Cluster. The other important thing is to organise the output files, as the pipeline generates a significant number of output files per genome and multiple genomes can be inputted into the pipeline at once. Folders for each tool's output are generated (Figure 7) alongside the data itself and then “wildcards” (a feature of Snakemake) ensure that the outputs from each tool are generated respectively, and provided with the correct file name.

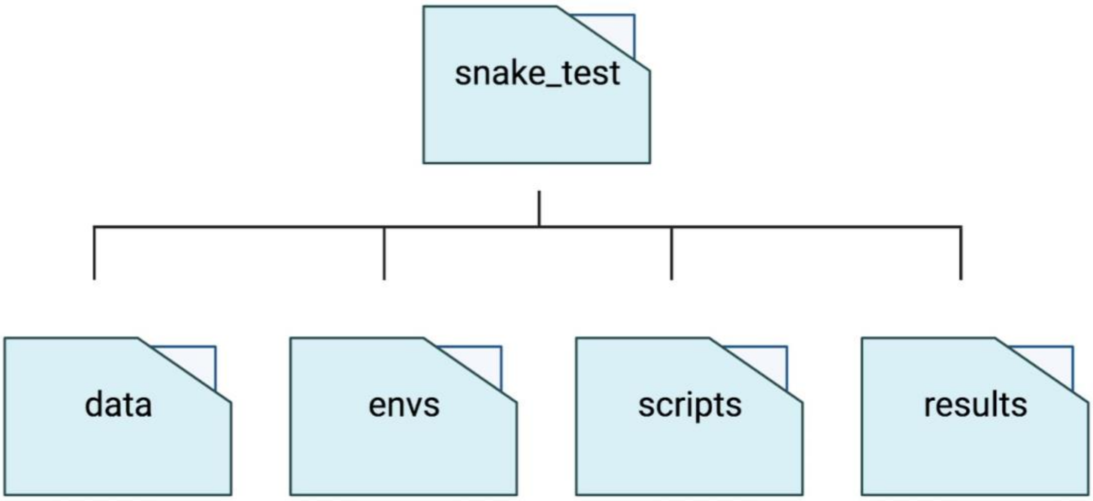


Figure 6. Schematic illustrating the initial file structure required to run the pipeline successfully.

Another vital part of setting up the pipeline was the installation of certain tools and databases. Yaml files, which contain the dependencies of Conda environment containers, were constructed for each tool of the tools in the pipeline, to ensure that the dependencies of one tool do not affect another.

```
snake_test
.
|---- Snakefile (.smk)
|---- data
|    |---- CLUSTER.fasta
|    |---- ncbi_data
|          |-- ecoli.fasta
|          |-- bsubtilis.fasta
|          |-- paeruginosa.fasta
|          |-- saureus.fasta
|          |-- vcholerae.fasta
|---- envs
|    |---- prokka_env.yaml
|    |---- pip_egg.yaml
|    |---- kofamscan.yaml
|    |---- keggdecoder.yaml |
|---- r_env.yaml
|    |---- seqkit_env.yaml
|    |---- kraken_env.yaml
|    |---- interproscan.yaml
|   |---- blast_env.yaml
|---- scripts
|   |---- egg_regex.py
|   |---- r_plot2.R
|   |---- stats_python.py
|---- results
|   |---- prokka_out_{wildcard}
|       |-- {wildcard}.faa
|       |-- {wildcard}.fna |
|-- {wildcard}.log
|       |-- {wildcard}.gff
|       |-- {wildcard}.ffn
|       |-- {wildcard}.fsa
|       |-- {wildcard}.tbl
|       |-- {wildcard}.sqn
|       |-- {wildcard}.gbk
|       |-- {wildcard}.txt
|   |---- egg_out_{wildcard}
|       |-- {wildcard}.emapper.annotations
|       |-- {wildcard}.emapper.hits
|       |-- {wildcard}.emapper.seed_orthologs
|   |---- kofamscan_out
|       |-- kofamscan_{wildcard}.tsv
|       |-- kofamscan_{wildcard}.txt
|   |---- decoder_out
|       |-- concat_{wildcard}
|       |-- decoder_out_{wildcard}
|   |---- seqkit_out
|       |-- hypothetical_list_{wildcard}
|       |-- seqkit_{wildcard}_length
|   |---- blast_out
|       |-- blast_{wildcard}
|   |---- interproscan_out
|       |-- {wildcard}_interpro.xml
|       |-- {wildcard}_interpro.gff3
|       |-- {wildcard}_interpro.tsv
|       |-- {wildcard}_interpro.xml
|   |---- kraken_2
|       |-- kraken_out_{wildcard}
|   |---- python_out
|       |-- {wildcard}_py
|         |-- subset
|         |-- stats_df
|         |-- low_qual_df
|         |-- egg_tax_df
|         |-- egg_module_df
|         |-- egg_cog_df
|   |---- r_plots
|       |-- {wildcard}_r
|         |-- plot1
|         |-- plot2
|         |-- plot3
|         |-- plot4
|        |-- plot5
|        |-- plot6
L        |-- plot_con
```

Figure 7. Schematic illustrating the file structure required for the pipeline, along with the expected output files that each tool will generate.

3.3 Installation

A High Performance Computing (HPC) cluster, Viking , is the cluster that is widely available to staff and researchers at the University of York. Viking was utilised to run command line based-tools and the develop the pipeline using Snakemake.

3.3.1 Protein Identification

3.3.1.1 PROKKA (1.14.5)

The modules ‘barrnap’ and ‘prokka’ were loaded in the Viking Cluster. The FASTA sequence of the MAG was then inputted into PROKKA to produce annotated output files. One of these output files includes the amino acid sequence file (*.faa) that was provided to run most of the bioinformatic tools in this pipeline.

3.3.1.2 EggNOG-mapper (v2.1.7)

A Conda environment was created, with EggNOG installed in the environment, in an appropriate workspace on the Viking Cluster, and then activated. The EggNOG database was then installed onto the Viking Cluster. The amino acid sequence of the MAG of interest, obtained from PROKKA, was used to run EggNOG producing a comma-separated variable output file.

3.3.1.3 KofamScan (1.3.0)

KofamScan was installed, via Unix command ‘wget’, to download the appropriate database and the tool onto the Viking cluster. These files were then unzipped using the ‘gzip’ Unix command. A conda environment was created with ‘kofamscan hmmer parallel’ and ‘ruby’ installed and then activated. A yaml file was made consisting of paths to find the databases in order to run KofamScan. The FASTA sequence and amino acid sequence (of the MAG) were required to run KofamScan. The tool produces two outputs, one tab-separated variable (tsv), and another that can be directly inputted into KEGG-Decoder.

3.3.1.4 KEGG-Decoder (V1.3)

A Conda environment was created with KEGG-Decoder and Python (version 3.6) installed in the enviroment and then activated. KEGG-Decoder was also installed in Python3 using ‘pip’, the output from KofamScan was then inputted into KEGG-Decoder and was run on the Viking cluster. The EggNOG output file and KofamScan output file were concatenated and inputted into KEGG-Decoder. A heatmap was outputted by KEGG-Decoder.

3.3.1.5 Python (3.9.7)

The EggNOG and KofamScan output files were read into Spyder and the module 'Pandas' was then read in. Basic information about both output files was obtained. Spyder was also used to make a new data frame of the EggNOG output file, to be an appropriate format for the KEGG-Decoder tool. Finally both output files were combined by an ‘outer join’ using the ‘merge()’ function. This new data frame was used for the confidence ranking system.

3.3.1.6 R Studio (4.1.2)

The EggNOG output file was read into a new file in R Studio, then the ‘ggplot2’ package was loaded in. Ggplot2 was used to provide plots of the data generated from the EggNOG output.

3.3.2 Confidence ranking system

3.3.2.1 Python (3.9.7)

The module ‘ttg’ in Python was used to generate simple truth tables that determine the logic for the confidence ranking system. An ‘elif’ loop was used to produce the confidence ranking system.

3.3.2.2 Microsoft Excel

Manual truth tables were manually generated using Microsoft Excel, that were more appropriate for the confidence values required for this dataset.

3.3.2.3 R studio (4.1.2)

Subset.csv was read into R studio. Then the ‘ggplot2’ package was loaded in. Values in the dataset were re-set as factors using ‘as.factors()’ in order to produce a number of plots visualising the results of the confidence ranking system.

3.3.3 Manual annotation

3.3.3.1 Kraken2 (2.1.2)

A Conda environment was created with Kraken2 installed. The Conda environment was then activated. Kraken2 was then run, with the amino acid sequence of the ORFs with a confidence value of 4, 5, or 6, producing a tab-separated variable output file.

3.3.3.2 NCBI protein BLAST+ (2.11.0-gompi-2021a)

Blast+ was loaded into the Viking cluster. The amino acid sequence of ORFs with a confidence value of 4, 5, or 6, was inputted and BLAST was run using the ‘outmft -6’ which produced a tsv file of the output results.

3.3.3.3 InterProScan (5.56-89.0)

Java 13.2.0 and then InterProScan 5.46-81.0-foss-2019b was loaded into the Viking cluster, as these were previously installed on viking. The amino acid sequence of the ORF of interest was input into InterProScan and a tab-separated variable table was outputted.

3.3.3.4 SeqKit (2.2.0)

The Conda environment ‘seqkit_env’ was created with SeqKit installed. This environment was then activated. A text file with the ORFs that were given a confidence value of 4, 5, or 6 , was inputted into SeqKit. The command ‘Seqkit grep -f’ was run, with the text file and the MAG amino acid sequence file inputted, to find the amino acid sequences of these individual ORFs. This SeqKit output file was then run again using the command ‘grep ‘hypothetical’ to list the number of ORFs that were annotated as hypothetical proteins. SeqKit was then run a final time using the command ‘fx2tab -l’ to count the sequence length of each ORF.

3.3.3.5 Snakemake (7.21.0)

Snakemake was installed via a Conda environment and was made using a .yaml file. Snakemake was run using a Snakefile file (*.smk), on unix command line. The command ‘snakemake’, along with a number of flags and parameters, was used to run the script written within the Snakefile.

3.4 Snakemake rules

Snakemake is a workflow management tool based in Python, that allows for the automation of this genome annotation pipeline. The pipeline consists of 18 rules, each of which will run a part of the pipeline. (Table 2) The first rule of the pipeline is called ‘rule all’.

Table 2. A summary of the outputs generated by each Snakemake rule in the pipeline.

snakemake rules	output
rule all	-
prokka	Table 9
eggnog	Table 10
egg_zero	Table 3
kofamscan	Table 11, 12
egg_for_decoder	Table 14
egg_format_decoder2	Table 4
rename_egg_decoder	Table 5
rename_kofam_decoder	Table 6
concat_for_decoder	Table 7
keggdecoder	Figure 9, 14
r_plots	Figure 10,11,12,13,15,16,17,18,19,20,21,22,23
python_info	Table 20, 18
low_qual	Table 8
kraken2	Table 21
interproscan	Table 22
blast_p	-
hypo_list	Table 23
seqkit_length	Table 24

‘Rule all’ is a rule that is used to direct the pipeline to run all of the rules, otherwise snakemake would only run the first rule in the pipeline. The input of this rule will always be the outputs of the final rule. Which in this case is called ‘seqkit_length’, and no output would be given. The second rule, called ‘prokka’, will run the tool PROKKA. The initial data is inputted as a FASTA sequence of the MAG(s). A yaml file for PROKKA is also included in this rule. The next rule is ‘eggnog’. This runs the EggNOG tool using a yaml file. The output of the ‘prokka’ rule (files ending with the suffix .faa) is also inputted into this tool. The output file of EggNOG places ‘-’ in the rows and columns that do not give a result. The ‘-’ will be changed to ‘0’ to allow for this output file to be compatible with the rules thereafter. This is made possible by the next rule called ‘egg_zero’, where a regular expression (regex) is run in shell to replace the ‘-’ for ‘0’ (Table 3). The rule ‘kofamscan’ then runs KofamScan. This also requires a yaml file to run KofamScan. Like eggnog, the output from the PROKKA rule is inputted into ‘kofamscan’. The outputs from KofamScan and EggNOG will later go on to be compared.

Table 3. Output file generated by the Snakemake rule 'egg_zero', all '-' replaced by '0' to be read correctly by the confidence system. The entire output file can be found in the Github repository, MRes-pipeline (https://github.com/louisegrimble/MRes-pipeline).

#query	seed_ortholog	evalue	score
JKJFLGDP_00001	1173027.Mic7113_1202	3.21E-85	275
JKJFLGDP_00002	756067.MicvaDRAFT_1231	4.63E-22	95.5
JKJFLGDP_00003	192952.MM_1348	0.000962	42.7
JKJFLGDP_00004	521011.Mpal_0288	3.16E-11	63.9
JKJFLGDP_00005	224325.AF_0894	2.55E-37	141
JKJFLGDP_00006	1094980.Mpsy_1669	1.30E-48	170
JKJFLGDP_00007	679926.Mpet_0612	3.29E-09	58.2
JKJFLGDP_00008	269797.Mbar_A1008	1.24E-14	75.5
JKJFLGDP_00009	1094980.Mpsy_1047	1.34E-26	105
JKJFLGDP_00010	1120936.KB907210_gene5880	1.00E-31	119

eggNOG_OGs
COG1506@1|root,COG1506@2|Bacteria,1G200@1117|Cyanobacteria,1H8T3@1150|Oscillatoriales
COG1506@1|root,COG1506@2|Bacteria,1G200@1117|Cyanobacteria,1H8T3@1150|Oscillatoriales
COG0018@1|root,arCOG00487@2157|Archaea,2XTWP@28890|Euryarchaeota,2N9C5@224756|Methanomicrobia
COG0018@1|root,arCOG00487@2157|Archaea,2XTWP@28890|Euryarchaeota,2N9C5@224756|Methanomicrobia
COG0018@1|root,arCOG00487@2157|Archaea,2XTWP@28890|Euryarchaeota,245YF@183980|Archaeoglobi
COG0018@1|root,arCOG00487@2157|Archaea,2XTWP@28890|Euryarchaeota,2N9C5@224756|Methanomicrobia
COG0018@1|root,arCOG00487@2157|Archaea,2XTWP@28890|Euryarchaeota,2N9C5@224756|Methanomicrobia
COG0018@1|root,arCOG00487@2157|Archaea,2XTWP@28890|Euryarchaeota,2N9C5@224756|Methanomicrobia
COG1842@1|root,arCOG04782@2157|Archaea,2XU11@28890|Euryarchaeota,2N9P1@224756|Methanomicrobia
COG1842@1|root,COG1842@2|Bacteria,2GP0K@201174|Actinobacteria,4EGYB@85012|Streptosporangiales

max_annot_lvl	COG_category	Description
1117|Cyanobacteria	E	PFAM Prolyl oligopeptidase family
1117|Cyanobacteria	E	PFAM Prolyl oligopeptidase family
224756|Methanomicrobia	J	Belongs to the class-I aminoacyl-tRNA synthetase family
224756|Methanomicrobia	J	Belongs to the class-I aminoacyl-tRNA synthetase family
183980|Archaeoglobi	J	Belongs to the class-I aminoacyl-tRNA synthetase family
224756|Methanomicrobia	J	Belongs to the class-I aminoacyl-tRNA synthetase family
224756|Methanomicrobia	J	Belongs to the class-I aminoacyl-tRNA synthetase family
224756|Methanomicrobia	J	Belongs to the class-I aminoacyl-tRNA synthetase family
224756|Methanomicrobia	K	PspA/IM30 family
201174|Actinobacteria	KT	PspA/IM30 family

Preferred_name	GOs	EC 0	KEGG_ko	KEGG_Pathway	KEGG_Module
dap2	0		0	0	0
dap2	0	0	0	0	0
argS	0	6.1.1.19	ko:K01887	ko00970,map00970	M00359,M00360
argS	0	6.1.1.19	ko:K01887	ko00970,map00970	M00359,M00360
argS	0	6.1.1.19	ko:K01887	ko00970,map00970	M00359,M00360
argS	0	6.1.1.19	ko:K01887	ko00970,map00970	M00359,M00360
argS	0	6.1.1.19	ko:K01887	ko00970,map00970	M00359,M00360
argS	0	6.1.1.19	ko:K01887	ko00970,map00970	M00359,M00360
0	0	0	ko:K03969	0	0
0	0	0	ko:K03969	0	0

KEGG_Reaction	KEGG_rclass	BRITE
0	0	0
0	0	0
R03646	RC00055,RC00523	ko00000,ko00001,ko00002,ko01000,ko01007,ko03016,ko03029
R03646	RC00055,RC00523	ko00000,ko00001,ko00002,ko01000,ko01007,ko03016,ko03029
R03646	RC00055,RC00523	ko00000,ko00001,ko00002,ko01000,ko01007,ko03016,ko03029
R03646	RC00055,RC00523	ko00000,ko00001,ko00002,ko01000,ko01007,ko03016,ko03029
R03646	RC00055,RC00523	ko00000,ko00001,ko00002,ko01000,ko01007,ko03016,ko03029
R03646	RC00055,RC00523	ko00000,ko00001,ko00002,ko01000,ko01007,ko03016,ko03029
0	0	ko00000
0	0	ko00000

KEGG_TC	CAZy	BiGG_Reaction	PFAMs
0	0	0	PD40,Peptidase_S9
0	0	0	PD40,Peptidase_S9
0	0	0	Arg_tRNA_synt_N,DALR_1,tRNA-synt_1d
0	0	0	Arg_tRNA_synt_N,DALR_1,tRNA-synt_1d
0	0	0	Arg_tRNA_synt_N,DALR_1,tRNA-synt_1d
0	0	0	Arg_tRNA_synt_N,DALR_1,tRNA-synt_1d
0	0	0	Arg_tRNA_synt_N,DALR_1,tRNA-synt_1d
0	0	0	Arg_tRNA_synt_N,DALR_1,tRNA-synt_1d
0	0	0	PspA_IM30
0	0	0	PspA_IM30

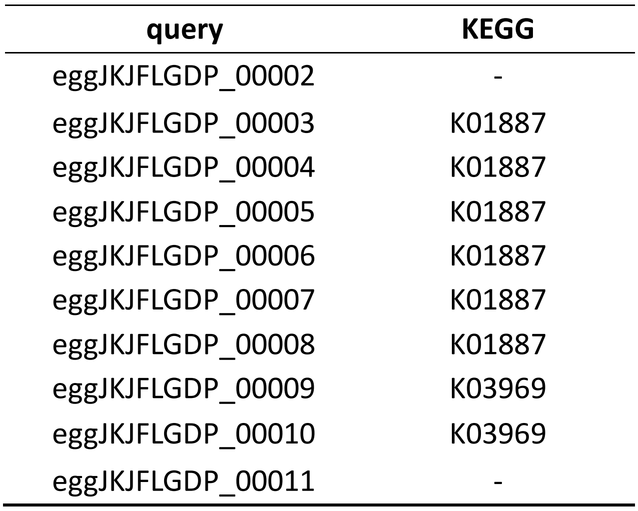
In order to input the EggNOG output into the tool, KEGG-Decoder, the output file needs to be reformatted by a python script in the rule ‘egg_for_decoder’. The python script removes the unwanted columns as well as removing some KEGG numbers from the column ‘KEGG_ko’. This is only necessary if that specific gene predicts more than one KEGG number. The final part of the script is a regex that removes the characters ‘ko:’ that precede the KEGG numbers in the column ‘KEGG_ko’ (Table 14). This is to make the format of the column the same as the output file of KofamScan. ‘egg_format_decoder’ is a rule that is also involved with reformatting the EggNOG output file to be inputted into KEGG-Decoder. The python script saves the output file as a comma-separated variable (.csv) however, this needs to be changed to a text file. The ‘,’ are removed from the file, and then the header is also removed (Table 4).

Table 4. Output file generated by Snakemake rule 'egg_format_decoder2', allowed the EggNOG output to be inputted into KEGG-Decoder to generate a heatmap. The entire output file can be found in the Github repository, MRes-pipeline (https://github.com/louisegrimble/MRes-pipeline).

query	KEGG
JKJFLGDP_00002	-
JKJFLGDP_00003	K01887
JKJFLGDP_00004	K01887
JKJFLGDP_00005	K01887
JKJFLGDP_00006	K01887
JKJFLGDP_00007	K01887
JKJFLGDP_00008	K01887
JKJFLGDP_00009	K03969
JKJFLGDP_00010	K03969
JKJFLGDP_00011	-

To determine the difference between the EggNOG annotation outputs and KofamScan annotation outputs, the name of the ORFs in the ‘query’ column need to differ. Two rules are used ensure this. ‘rename_egg_decoder’ adds the characters ‘egg’ to the start of every ORF from the EggNOG results (Table 5), and another rule called ‘rename_kofam_decoder’ adds the characters ‘kofam’ to the start of every ORF from the KofamScan results (Table 6).

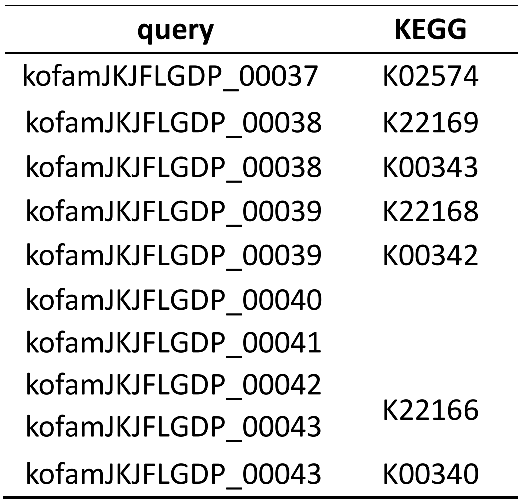
Table 5. Output file generated by Snakemake rule 'rename_egg_decoder', added ‘egg’ to each of the query sequences to identify which row in the heatmap was from the EggNOG output. The entire output file can be found in the Github repository, MRes-pipeline (https://github.com/louisegrimble/MRes-pipeline).



query	KEGG
eggJKJFLGDP_00002	-
eggJKJFLGDP_00003	K01887
eggJKJFLGDP_00004	K01887
eggJKJFLGDP_00005	K01887
eggJKJFLGDP_00006	K01887
eggJKJFLGDP_00007	K01887
eggJKJFLGDP_00008	K01887
eggJKJFLGDP_00009	K03969
eggJKJFLGDP_00010	K03969
eggJKJFLGDP_00011	-

The final rule required to reformat the EggNOG and KofamScan outputs is ‘concat_for_decoder’ (Table 7). This is predominantly due to KEGG-Decoder requiring two or more sequences to be inputted at once in order to generate a heatmap. This also ensures a single heatmap is generated per dataset that is passed through in the pipeline. Finally, KEGG-decoder will be run by the rule ‘keggdecoder’ using a yaml file of KEGG-Decoder. The output from ‘concat_for_decoder’ is inputted into this rule and a heatmap is generated as an output, as an svg.

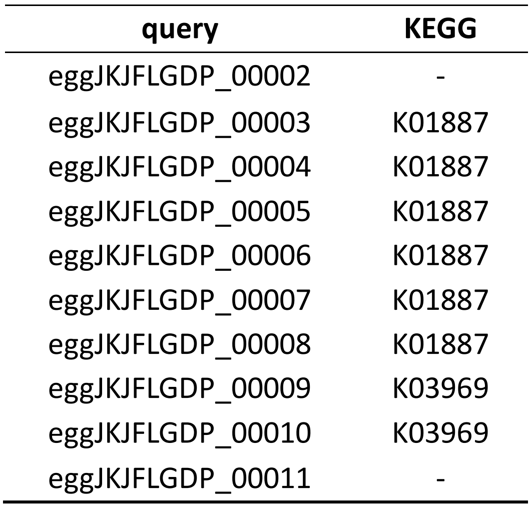
Table 6. Output file generated by Snakemake rule 'rename_kofam_decoder', added ‘kofam’ to each of the query sequences to identify which row in the heatmap was from the KofamScan output. The entire output file can be found in the Github repository, MRespipeline (https://github.com/louisegrimble/MRes-pipeline).



query	KEGG
kofamJKJFLGDP_00037	K02574
kofamJKJFLGDP_00038	K22169
kofamJKJFLGDP_00038	K00343
kofamJKJFLGDP_00039	K22168
kofamJKJFLGDP_00039	K00342
kofamJKJFLGDP_00040	
kofamJKJFLGDP_00041	
kofamJKJFLGDP_00042	K22166
kofamJKJFLGDP_00043	
kofamJKJFLGDP_00043	K00340

The next rule is ‘python_info’, this consists of a python script that summarises the output files from EggNOG and KofamScan, and will run the confidence ranking system of the annotations from EggNOG and KofamScan. A summary table, a table of low quality ORFs (determined by the confidence system), and other tables (Supplementary data S.1), are outputted from this rule. The rule ‘r_plots’ uses R to generate a number of plots, from the EggNOG output data, and the confidence ranking system output data.

Table 7. Output file generated by Snakemake rule 'concat_for_decoder', Table 5. and Table 6. were concatenated (as axis=0) to be inputted into KEGG-Decoder. The entire output file can be found in the Github repository, MRes-pipeline (https://github.com/louisegrimble/MRes-pipeline).



query	KEGG
eggJKJFLGDP_00002	-
eggJKJFLGDP_00003	K01887
eggJKJFLGDP_00004	K01887
eggJKJFLGDP_00005	K01887
eggJKJFLGDP_00006	K01887
eggJKJFLGDP_00007	K01887
eggJKJFLGDP_00008	K01887
eggJKJFLGDP_00009	K03969
eggJKJFLGDP_00010	K03969
eggJKJFLGDP_00011	-

The table of low quality ORFs from the ‘python_info’ rule is inputted into the next rule ‘low_qual’. Seqkit is used along with grep to select all of the amino acid sequences of the low quality ORFs, this file will allow the low quality ORFs to undergo through further annotation. Next, the low quality ORF amino acid file from the previous rule (‘low_qual’) is inputted into the rule ‘kraken’. This rule will involve running the tool Kraken2, using a yaml file to run the tool. The low quality ORF amino acid file (Table 8) is also inputted into the next two rules ‘interproscan’, which runs the tool InterProScan and ‘blast_p’, which runs the protein BLAST search tool. Both rules (interproscan and blast_p) will use a yaml file, respectively, of the tool to run them. The next rule is ‘hypo_list’, this rule will generate a file of every ORF that has been defined by PROKKA as a ‘hypothetical’ protein. These ORFs will not typically provide any results to the tools used prior.

Table 8. Output file generated by snakemake rule 'low_qual', SeqKit tool collate the amino acid sequences of each of the low quality ORFs (Table 20.) from the confidence system. The entire output file can be found in the Github repository, MRes-pipeline (https://github.com/louisegrimble/MRes-pipeline).

```
>JKJFLGDP_00001 hypothetical protein
MSWFQPLQCQDPVHEGGGAYTVFGRTLIFSNFADQRLYRLDPARPPAHHPAIECRYADGV
INARCRPITCACEDHRLPGREALNTIVAIDLDGEVEMQVLAKGYDFYPSPRISPDGRRLA
WPAWHHPQMPWDGTELWTGELEKDGSLGAVERVAGGSDESIFQPQWSPDGILHFVSDRTG
WSNLYCWQDGHVLALTDIQAELSRPQWRFGFSTYAFLTPDRIICTYAQDGIWKLARLDTS
YLKIDPIKTPFTEISYLEACRIMPSSSPALLIWPPRW
>JKJFLGDP_00002 hypothetical protein
MQKLAPYGSWRSPITSDMIASQTIGLEQIALEGSDIYWIESRPAEGGRSVIIRCDPGGGV
DELVPAPSMPGPGA
```

The final rule of the pipeline is ‘seqkit_length’, this will retrieve the sequence length of each ORF. This rule combined with the penultimate rule, ‘hypo_list’, can determine whether the ORFs are potentially incomplete proteins. If the ORFs are both hypothetical and smaller than typical gene length for prokaryotes, they can be disregarded as a result of poor sequencing.

Chapter 4: Results (*Methanothrix soehngenii*)

4.1 Overall pipeline

The pipeline was split into 3 main steps (Figure 8). The first step of the pipeline, protein identification, was to produce a functional annotation. This annotation involved providing a KEGG number to every ORF that was recognised by the reference databases of the tools. The second step of the pipeline, confidence system, involved a confidence ranking system that was devised based on the completeness of the annotations generated in step 1. In the final step of the pipeline, manual annotation, involved low quality ORFs (determined by the confidence system) that were placed into a new dataframe. This dataframe was passed through a series of functional and taxonomic annotations. This provided more breadth to the initial annotations generated by comparison of the sequences to a larger number of reference databases. The sequence lengths of these ORFs was discovered from this which provided greater understanding of why the proteins were unknown e.g. they were incomplete proteins during sequencing or perhaps subject to truncation via mutations.

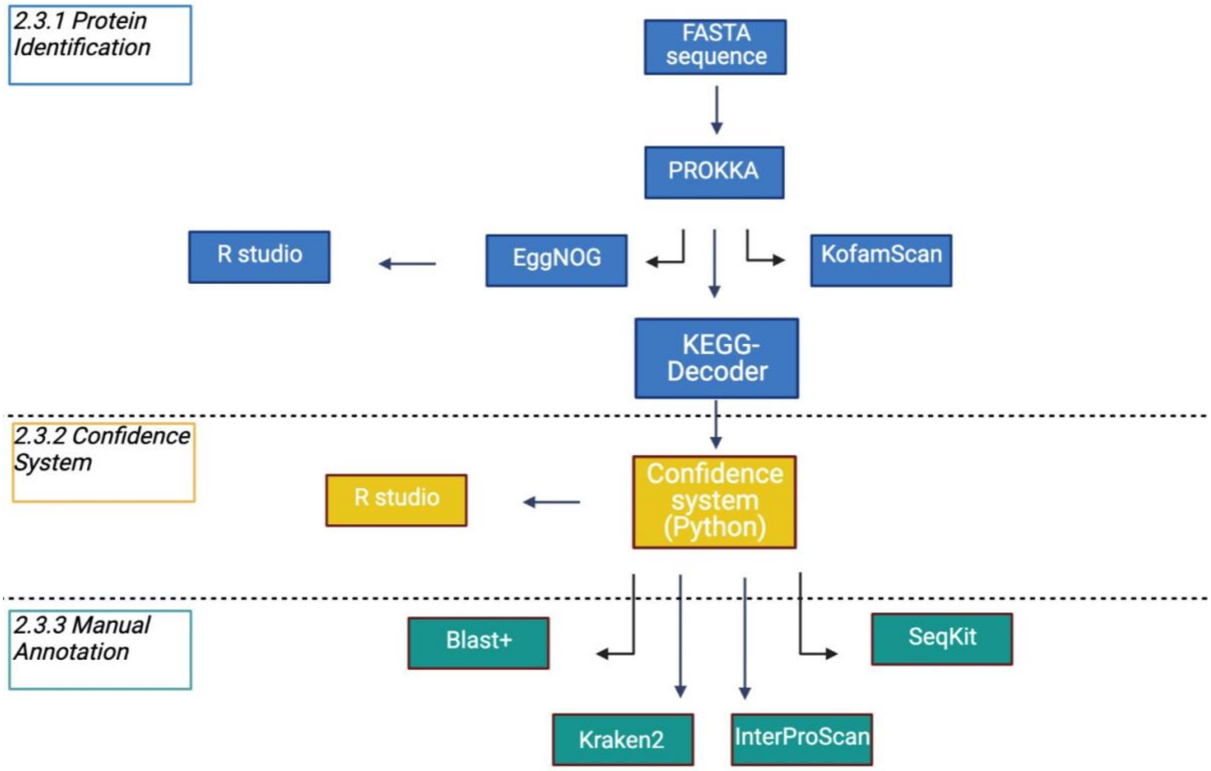


Figure 8. Directed acyclic graph illustrating the three main steps of the functional annotation pipeline. The main steps are 2.3.1 Protein Identification (blue), 2.3.2 Confidence Ranking (yellow) and 2.3.3 Manual Annotation (green).

4.2 PROKKA output for Methanothrix soehngenii

The first output produced by the pipeline is the PROKKA output files. The amino acid sequence file (*.faa) is utilised as an input to run many of the bioinformatic tools in this pipeline (Table 9).

Table 9. Output file generated by PROKKA for the Methanothrix soehngenii MAG. Amino acid sequence of each ORF in this MAG. The entire output file can be found in the Github repository, MRes-pipeline (https://github.com/louisegrimble/MRes-pipeline).

```
>JKJFLGDP_00001 hypothetical protein
MSWFQPLQCQDPVHEGGGAYTVFGRTLIFSNFADQRLYRLDPARPPAHHPAIECRYADGV
INARCRPITCACEDHRLPGREALNTIVAIDLDGEVEMQVLAKGYDFYPSPRISPDGRRLA
WPAWHHPQMPWDGTELWTGELEKDGSLGAVERVAGGSDESIFQPQWSPDGILHFVSDRTG
WSNLYCWQDGHVLALTDIQAELSRPQWRFGFSTYAFLTPDRIICTYAQDGIWKLARLDTS
YLKIDPIKTPFTEISYLEACRIMPSSSPALLIWPPRW
>JKJFLGDP_00005 Arginine--tRNA ligase
MIGDTLVRILRRAGCTVDAQYYINDMGRQEAMVVVGCDHFQLDDSKADHATARVYIAANK
EMETTPAIREEADRVIQLYEAGDPKITAKIQSAVRYAISGIEETLERMNIRHDNYHWESE
FVRDGSVAEILKRLEDRPGGLGGGIASAGSERIRF
```

4.3 EggNOG output for Methanothrix soehngenii

The EggNOG output file provides each ORF with an E-value, predicted protein name, KEGG number, functional COG, along with other information (Table 10). The E-value is defined by the estimated probability that the protein will be found in the database. Therefore, the closer the E-value is to zero, the better the prediction of the ORF. The KEGG KO is an identifier given to each protein that is recognised by the KEGG Orthology. Which can then be used to refer to KEGG pathways, to identify the role and/or function of the protein in certain biological processes. Functional Cluster of Orthologous Groups (COGs) is the classification of proteins in order to try to determine their function via phylogenetic analysis. The KEGG KO and COG predictions will later determine the confidence values of the confidence ranking system. However, not all of this information from the EggNOG annotation is always provided to each ORF. This could be due to a number of factors including the query sequence not being present in the KEGG database, or the protein has not been discovered yet (often referred to as a hypothetical protein). There also could have been errors during sequencing. For example, the use of different techniques can provide varied results, as Illumina sequencing is more accurate than Nanopore sequencing (McNaughton, et al., 2019). Finally, recovering DNA from environmental samples is likely to result in errors (Mande, Mohammed and Ghosh, 2012).

Table 10. Output file generated by EggNOG for the Methanothrix soehngenii MAG. The entire output file can be found in the Github repository, MRes-pipeline (https://github.com/louisegrimble/MRes-pipeline).

#query	**seed_ortholog**	**evalue**	**score**
JKJFLGDP_00001	1173027.Mic7113_1202	3.21E-85	275
JKJFLGDP_00002	756067.MicvaDRAFT_1231	4.63E-22	95.5
JKJFLGDP_00003	192952.MM_1348	0.000962	42.7
JKJFLGDP_00004	521011.Mpal_0288	3.16E-11	63.9
JKJFLGDP_00005	224325.AF_0894	2.55E-37	141
JKJFLGDP_00006	1094980.Mpsy_1669	1.30E-48	170
JKJFLGDP_00007	679926.Mpet_0612	3.29E-09	58.2
JKJFLGDP_00008	269797.Mbar_A1008	1.24E-14	75.5
JKJFLGDP_00009	1094980.Mpsy_1047	1.34E-26	105
JKJFLGDP_00010	1120936.KB907210_gene5880	1.00E-31	119

eggNOG_OGs
COG1506@1|root,COG1506@2|Bacteria,1G200@1117|Cyanobacteria,1H8T3@1150|Oscillatoriales
COG1506@1|root,COG1506@2|Bacteria,1G200@1117|Cyanobacteria,1H8T3@1150|Oscillatoriales
COG0018@1|root,arCOG00487@2157|Archaea,2XTWP@28890|Euryarchaeota,2N9C5@224756|Methanomicrobia
COG0018@1|root,arCOG00487@2157|Archaea,2XTWP@28890|Euryarchaeota,2N9C5@224756|Methanomicrobia
COG0018@1|root,arCOG00487@2157|Archaea,2XTWP@28890|Euryarchaeota,245YF@183980|Archaeoglobi
COG0018@1|root,arCOG00487@2157|Archaea,2XTWP@28890|Euryarchaeota,2N9C5@224756|Methanomicrobia
COG0018@1|root,arCOG00487@2157|Archaea,2XTWP@28890|Euryarchaeota,2N9C5@224756|Methanomicrobia
COG0018@1|root,arCOG00487@2157|Archaea,2XTWP@28890|Euryarchaeota,2N9C5@224756|Methanomicrobia
COG1842@1|root,arCOG04782@2157|Archaea,2XU11@28890|Euryarchaeota,2N9P1@224756|Methanomicrobia
COG1842@1|root,COG1842@2|Bacteria,2GP0K@201174|Actinobacteria,4EGYB@85012|Streptosporangiales

max_annot_lvl	COG_category	Description
1117|Cyanobacteria	E	PFAM Prolyl oligopeptidase family
1117|Cyanobacteria	E	PFAM Prolyl oligopeptidase family
224756|Methanomicrobia	J	Belongs to the class-I aminoacyl-tRNA synthetase family
224756|Methanomicrobia	J	Belongs to the class-I aminoacyl-tRNA synthetase family
183980|Archaeoglobi	J	Belongs to the class-I aminoacyl-tRNA synthetase family
224756|Methanomicrobia	J	Belongs to the class-I aminoacyl-tRNA synthetase family
224756|Methanomicrobia	J	Belongs to the class-I aminoacyl-tRNA synthetase family
224756|Methanomicrobia	J	Belongs to the class-I aminoacyl-tRNA synthetase family
224756|Methanomicrobia	K	PspA/IM30 family
201174|Actinobacteria	KT	PspA/IM30 family

Preferred_name	GOs	EC	KEGG_ko	KEGG_Pathway	KEGG_Module
dap2	-	-	-	-	-
dap2	-	-	-	-	-
argS	-	6.1.1.19	ko:K01887	ko00970,map00970	M00359,M00360
argS	-	6.1.1.19	ko:K01887	ko00970,map00970	M00359,M00360
argS	-	6.1.1.19	ko:K01887	ko00970,map00970	M00359,M00360
argS	-	6.1.1.19	ko:K01887	ko00970,map00970	M00359,M00360
argS	-	6.1.1.19	ko:K01887	ko00970,map00970	M00359,M00360
argS	-	6.1.1.19	ko:K01887	ko00970,map00970	M00359,M00360
-	-	-	ko:K03969	-	-
-	-	-	ko:K03969	-	-

KEGG_Reaction	KEGG_rclass	BRITE
-	-	-
-	-	-
R03646	RC00055,RC00523	ko00000,ko00001,ko00002,ko01000,ko01007,ko03016,ko03029
R03646	RC00055,RC00523	ko00000,ko00001,ko00002,ko01000,ko01007,ko03016,ko03029
R03646	RC00055,RC00523	ko00000,ko00001,ko00002,ko01000,ko01007,ko03016,ko03029
R03646	RC00055,RC00523	ko00000,ko00001,ko00002,ko01000,ko01007,ko03016,ko03029
R03646	RC00055,RC00523	ko00000,ko00001,ko00002,ko01000,ko01007,ko03016,ko03029
R03646	RC00055,RC00523	ko00000,ko00001,ko00002,ko01000,ko01007,ko03016,ko03029
-	-	ko00000
-	-	ko00000

KEGG_TC	CAZy	BiGG_Reaction	PFAMs
-	-	-	PD40,Peptidase_S9
-	-	-	PD40,Peptidase_S9
-	-	-	Arg_tRNA_synt_N,DALR_1,tRNA-synt_1d
-	-	-	Arg_tRNA_synt_N,DALR_1,tRNA-synt_1d
-	-	-	Arg_tRNA_synt_N,DALR_1,tRNA-synt_1d
-	-	-	Arg_tRNA_synt_N,DALR_1,tRNA-synt_1d
-	-	-	Arg_tRNA_synt_N,DALR_1,tRNA-synt_1d
-	-	-	Arg_tRNA_synt_N,DALR_1,tRNA-synt_1d
-	-	-	PspA_IM30
-	-	-	PspA_IM30

4.4 KofamScan output for Methanothrix soehngenii

The KofamScan output file (Table 11), similar to that of EggNOG, provides each ORF with an E-value and KEGG number. KofamScan produces three different results per ORF, rather than one result per ORF in the EggNOG annotation. To account for this, the repeats with the lowest E-value were removed to ensure the reproducibility of the confidence system. As mentioned previously, the KEGG KO from KofamScan will also be involved in the determination of confidence values.

Table 11. Output file generated by KofamScan for the Methanothrix soehngenii MAG. The tsv file was generated to allow for further analysis in the confidence system. The entire output file can be found in the Github repository, MRes-pipeline (https://github.com/louisegrimble/MRes-pipeline).

gene name	KO	thrshld	score	E-value	KO definition
---------	------	-------	------	---------	-------------
JKJFLGDP_00001	K03641	120.57	39.8	5.40E-11	TolB protein
JKJFLGDP_00001	K08676	573.57	23	1.90E-06	tricorn protease [EC:3.4.21.-]
JKJFLGDP_00001	K19732	171.77	11	0.022	LuxR family transcriptional regulator, activator of conjugal transfer of Ti plasmids
JKJFLGDP_00003	K13077	762.57	10.5	0.027	flavone synthase I [EC:1.14.20.5]
JKJFLGDP_00004	K01887	127.03	41.8	8.10E-12	arginyl-tRNA synthetase [EC:6.1.1.19]
JKJFLGDP_00004	K01886	508.87	14.8	0.00089	glutaminyl-tRNA synthetase [EC:6.1.1.18]
JKJFLGDP_00005	K01887	127.03	96.2	3.00E-28	arginyl-tRNA synthetase [EC:6.1.1.19]
JKJFLGDP_00006	K01887	127.03	81.5	8.30E-24	arginyl-tRNA synthetase [EC:6.1.1.19]
JKJFLGDP_00006	K04566	231.2	17.8	9.30E-05	lysyl-tRNA synthetase, class I [EC:6.1.1.6]
JKJFLGDP_00007	K01887	127.03	18.4	9.20E-05	arginyl-tRNA synthetase [EC:6.1.1.19]

KofamScan produces an output file format that can be directly inputted into KEGG-Decoder (Table 12). This output consists of the ORF name and the KEGG KO that KofamScan has assigned to each of the ORFs.

Table 12. Output file generated by KofamScan for the Methanothrix soehngenii MAG. The txt file was generated to be directly inputted into KEGG-Decoder to generate a heatmap. The entire output file can be found in the Github repository, MRes-pipeline (https://github.com/louisegrimble/MRes-pipeline).

query	KEGG
JKJFLGDP_00037	K02574
JKJFLGDP_00038	K22169
JKJFLGDP_00038	K00343
JKJFLGDP_00039	K22168
JKJFLGDP_00039	K00342
JKJFLGDP_00040	
JKJFLGDP_00041	
JKJFLGDP_00042	K22166
JKJFLGDP_00043	
JKJFLGDP_00043	K00340

4.5 Summary of EggNOG and KofamScan annotation for Methanothrix soehngenii

A series of statistics were generated, to compare both EggNOG and KofamScan outputs (Table 13). EggNOG and KofamScan produce differing predictions for the same query sequence i.e. they identify a different total number of proteins. EggNOG identifies 1980 proteins and KofamScan identifies 2098 proteins. This difference could be due to the variation between the profile hidden Markov models used by KofamScan and the KEGG orthology database used by EggNOG. 31.46% of the proteins annotated by EggNOG were not assigned a KEGG KO. Whereas, all 2098 proteins that were annotated by KofamScan are assigned a KEGG KO. KofamScan will seemingly provide a prediction regardless of the quality, therefore, predictions made by KofamScan are likely to be less precise than EggNOG. The application of a confidence measure will ensure the results generated are accurate and will automatically remove ORFs that were not provided an accurate prediction.

Table 13. Summary of the EggNOG and KofamScan annotations (Methanothrix soehngenii MAG).

Proteins predicted EggNOG	2103
Proteins predicted KofamScan (incl. duplicates)	1099
Proteins predicted KofamScan	2099
Proteins without KEGG number (EggNOG)	804 (38.2%)
Proteins without KEGG number (KofamScan)	0 (0.0%)

4.6 Manipulation of EggNOG output file for input into KEGG-Decoder.

In order to input the EggNOG annotation into KEGG-Decoder, manipulation of the EggNOG output file format is required. For example, the EggNOG output file presents KEGG KOs as “ko:K00399” and KofamScan presents the same KEGG KO as “K00399”. Following a multiple step manipulation, a final file format (Table 14) of the EggNOG annotation output is generated.

Table 14. Output file generated by Snakemake rule 'egg_for_decoder', removed additional columns that are not required to run KEGG-Decoder. The entire output file can be found in the Github repository, MRes-pipeline (https://github.com/louisegrimble/MRes-pipeline).

,query,KEGG
0,JKJFLGDP_00002,-
1,JKJFLGDP_00003,K01887
2,JKJFLGDP_00004,K01887
3,JKJFLGDP_00005,K01887
4,JKJFLGDP_00006,K01887
5,JKJFLGDP_00007,K01887
6,JKJFLGDP_00008,K01887
7,JKJFLGDP_00009,K03969
8,JKJFLGDP_00010,K03969
9,JKJFLGDP_00011,-

4.7 KEGG-decoder illustrates the metabolism of Methanothrix soehngenii.

The heatmap (Figure 9) generated by KEGG-decoder visualises the number of genes that are found in a number of biological processes, according to the KEGG orthology. *Methanothrix soehngenii* is an acetoclastic methanogen which is illustrated by Figure 9. This confirms the predictions of protein function via EggNOG and KofamScan is to a high standard. The annotations provided by both tools were similar overall, with some exceptions such as Glucoamylase, and the rTCA cycle. In the rTCA cycle, EggNOG predicted a high frequency of proteins, and KofamScan predicted no proteins. In the Glucoamylase pathway EggNOG predicted no proteins and KofamScan predicted a high frequency of proteins. This is likely due to the reference databases of each tool consisting of different reference sequences, providing varied predictions. However, the EggNOG and KofamScan annotations of the metabolic pathways involved in acetoclastic methanogenesis e.g. Methanogenesis via acetate, methanogenesis via CO_2, coenzyme B/ coenzyme M regeneration and coenzyme M reduction to methane; are very similar. Both tools predicted high frequencies of proteins within these pathways, which is consistent with *M. soehngenii* being an acetoclastic methanogen.

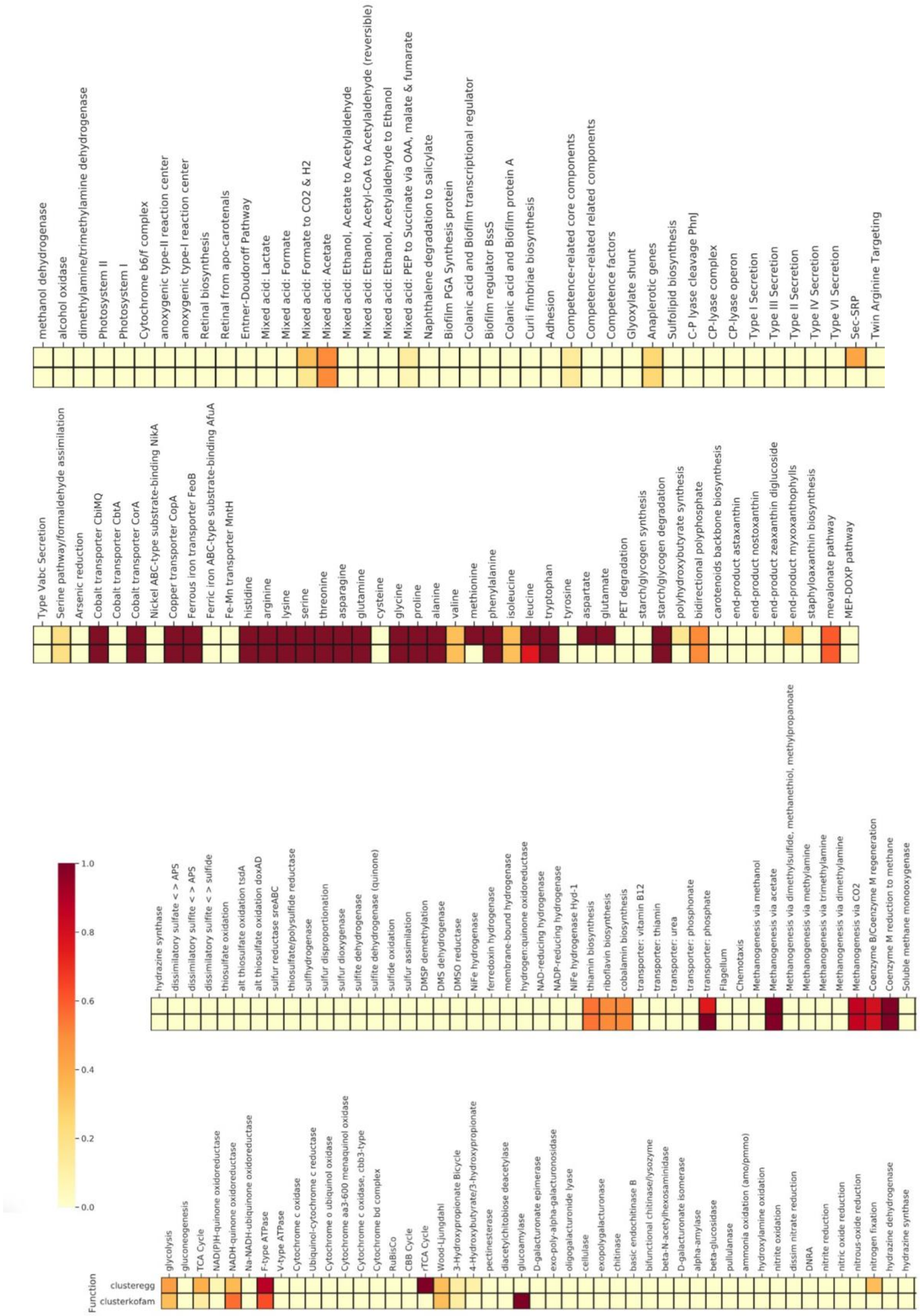


Figure 9. Heatmap generated by KEGG-Decoder illustrates the completeness of metabolic pathways in Methanothrix soehngenii MAG, recognised by KEGGmapper. Dark red - 1, white- 0, 1 is equivalent to complete pathway and 0 is equivalent to the entire pathway being absent.

4.8 Taxonomic analysis of Methanothrix soehngenii via EggNOG.

The taxonomic levels of each ORF within the *M. soehngenii* MAG was compared against the E-value prediction (Figure 10) from the EggNOG annotation. EggNOG predicted proteins from 3 methanogenic orders of the *Euryarchaeaota* phylum (*Methanomicrobia, Methanococcus, and Methanobacteria*). The E-value illustrates the confidence of the taxonomic classification of the ORFs.

Figure 3. Comparing taxonomy level with E-value

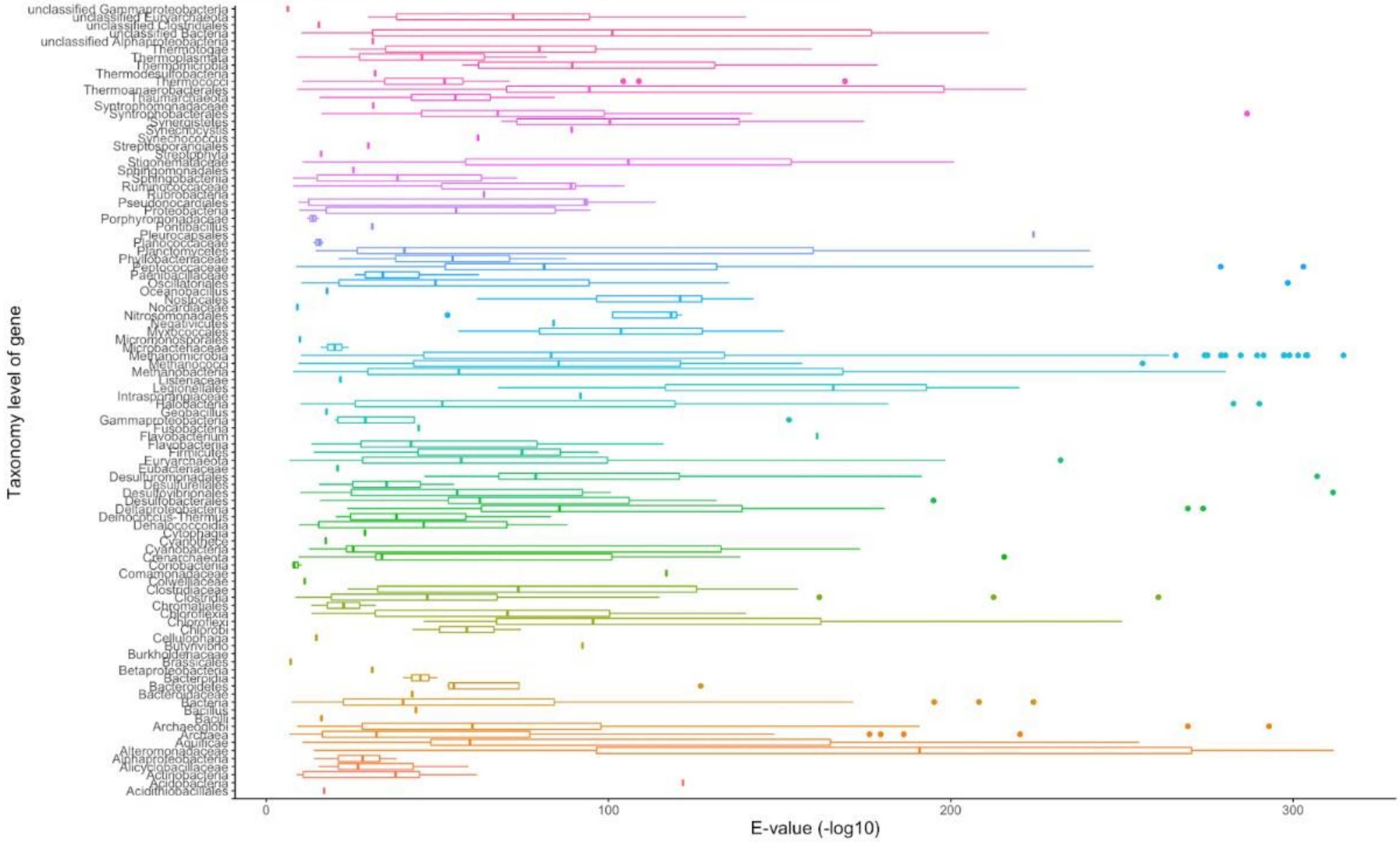
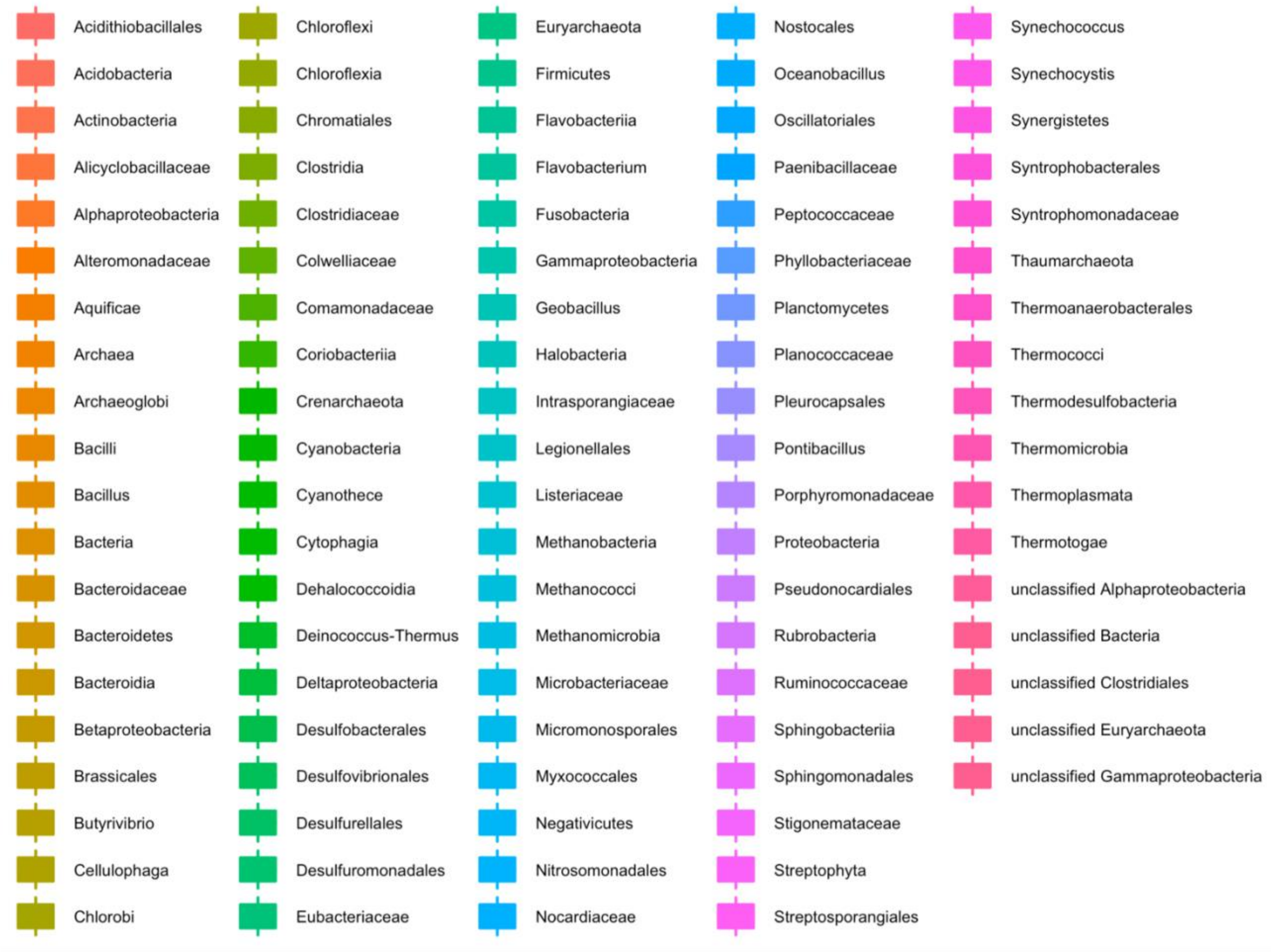


Figure 10. Boxplot illustrating the taxonomic analysis generated by EggNOG compared against the E-value of each protein prediction, for the Methanothrix soehngenii MAG. The plot was generated using ggplot2 in R studio and the data was log-transformed (-Log10). Colour legend:



4.9 E-value distribution of EggNOG protein prediction

Figure 11 illustrates the distribution of all E-values of EggNOG predictions, however the distribution is not clear with this scale. The difference between the E-values is so small that the difference is not visualised in this format. To combat this issue the data was transformed, by -log10, ensuring the distribution is clearly visible.

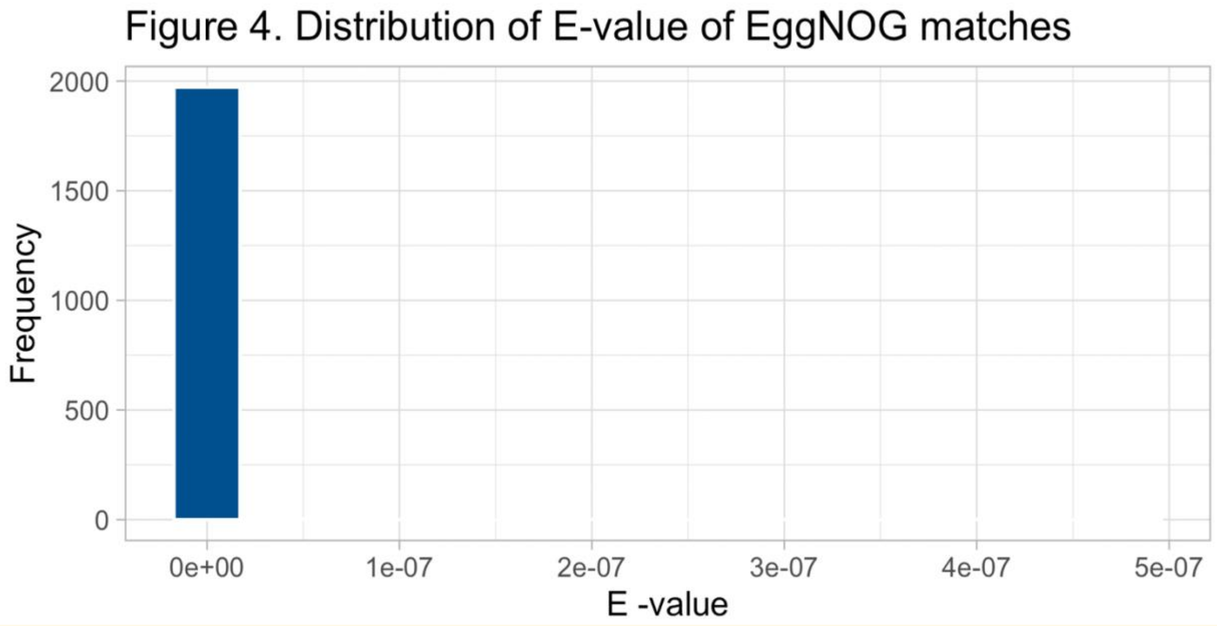


Figure 11. Histogram illustrating the E-values of the predictions made by EggNOG of the M. soehngenii MAG sequence. All predictions have a relatively good E-value as they are all close to zero, however this plot does not visualise the overall distribution.

The overall E-value distribution of predictions (of the *M. soehngenii* MAG) is represented as a histogram (Figure 12). This plot shows that the data is negatively skewed towards zero. Evalues closer to zero are desired, as it indicates that the predictions generated by EggNOG are reliable and the user can be confident that a high proportion of the proteins have been predicted accurately. However, this data has been transformed (-log10), meaning that the data shows that there are few predictions close to zero (or one in Figure 12) in proportion to the scale in Figure 12. There is a high number of E-values towards one (or zero in Figure 12), although the difference between all of the E-values in the sequences is very small, this can be seen clearly in Figure 11.

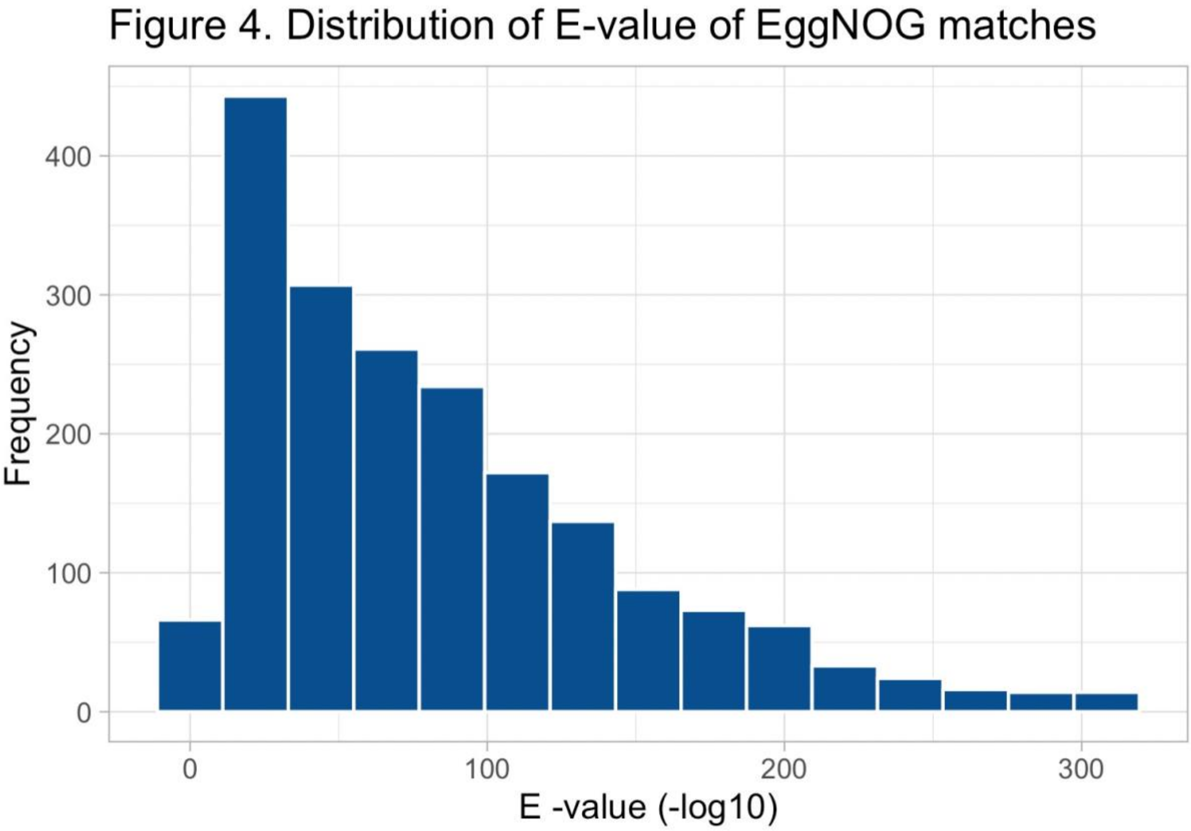


Figure 12. Histogram illustrating the E-values of the predictions made by EggNOG of the M. soehngenii MAG sequence. The data is skewed towards zero, however the data was log-transformed (-Log10), therefore there are only few E-values with an exceptional E-value.

4.10 Functional COGs predicted by EggNOG in Methanothrix soehngenii MAG.

The number of proteins assigned to each of the functional COGs (Cluster of Orthologous Groups) is visualised in a bar plot (Figure 13). This provides more information about the function of groups of proteins based on their taxonomy. Functional COGs can therefore help to determine the overall function of the organism. EggNOG predicted a high abundance of proteins belonging to S and L functional COGs in the *M. soehngenii* MAG. The S functional COG is defined as an ‘unknown function’, whereas L is responsible for replication, recombination and repair (Chen, Xia and Li, 2022). A high abundance of L COG is expected as they’re involved in basic cellular processes. COGs involved in metabolism also had a prominent number of proteins present. These COGs inculded P (inorganic ion transport and metabolism), H (Coenzyme transport and metabolism), and C (energy production and conversion). This result is expected due to the fact that the sole energy source of *M. soehngenii* is acetate and is metabolised by the acetoclastic methanogenesis pathway, which involves coenzyme M and and B (Horng, Becker and Ragsdale, 2001).

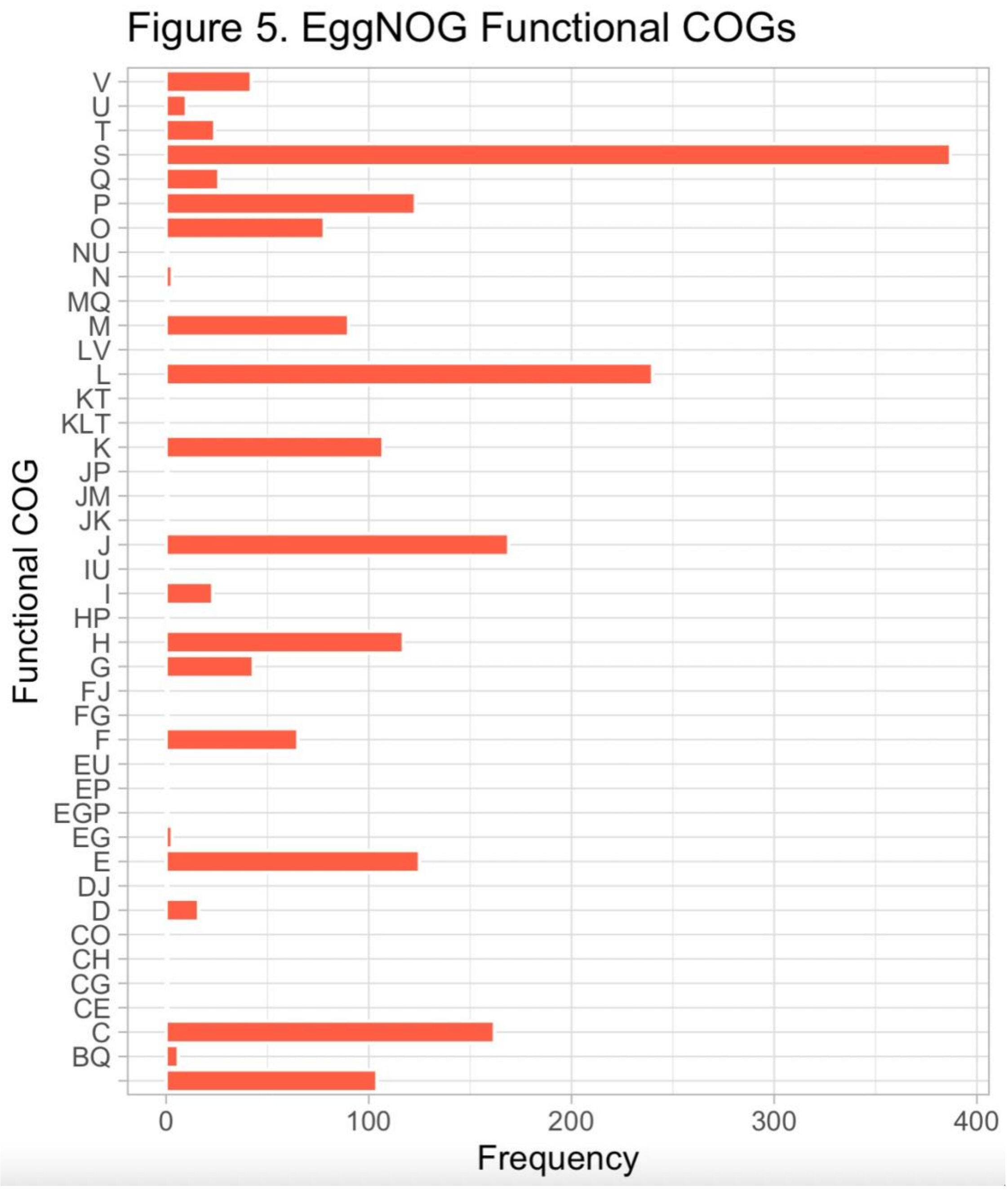
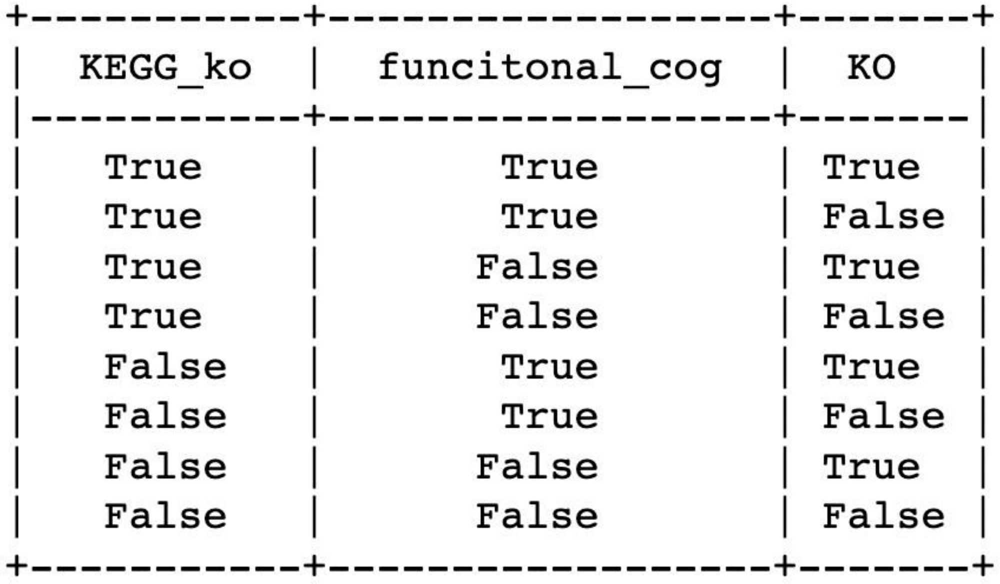


Figure 13. Bar plot illustrating the frequency of different functional Cluster of Orthologous Groups (COGs) predicted by EggNOG in the Methanothrix soehngenii MAG sequence. The majority of proteins predicted were assigned the S COG, which are described as proteins of unknown function.

4.11 Truth tables were generated to devise confidence ranking system

A Truth table (Table 15) was generated to devise the logic for the confidence ranking system. The tables compare the functional COG from the EggNOG output, the KEGG KO from EggNOG and the KEGG KO from KofamScan. This truth table is not entirely applicable to the dataset as there is an additional confidence value that the automatically generated truth table cannot determine.

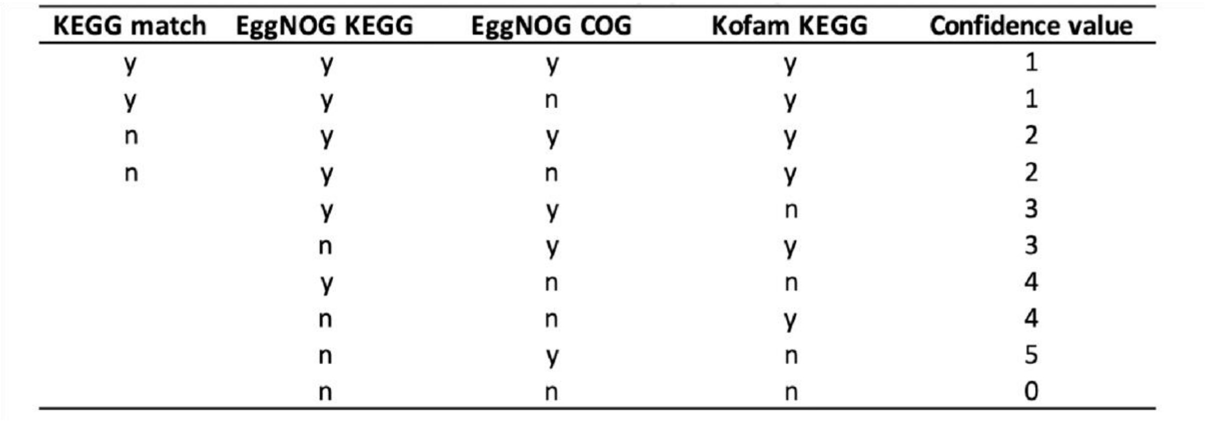
Table 15. Truth table generated by Python module ‘ttg’, logic of truth table used to develop the confidence rank system.



```
+-----------+------------------+-------+
| KEGG_ko   | funcitonal_cog   | KO    |
|-----------+------------------+-------|
|   True    |       True       | True  |
|   True    |       True       | False |
|   True    |      False       | True  |
|   True    |      False       | False |
|   False   |       True       | True  |
|   False   |       True       | False |
|   False   |      False       | True  |
|   False   |      False       | False |
+-----------+------------------+-------+
```

A manually-made truth table (Table 16) was generated to include the additional confidence value (confidence value of 1). This value is assigned to ORFs with the highest confidence in the annotations. Confidence value 1 is defined as when the KEGG KO from both EggNOG and KofamScan, were predicted to be the same value, and a functional COG was predicted. The confidence value 2 is defined as a ORF that was predicted a KEGG KO from both tools, but that KEGG KO was not identical. A confidence value of 3 is appointed when either EggNOG or KofamScan predicted a KEGG KO (but not both) and a functional COG is provided. A confidence value of 4 is given when either EggNOG or KofamScan provide a KEGG number but no functional COG is provided. Confidence value of 5 is given when only a functional COG is provided by EggNOG and finally, a confidence value of 6 is a ORF that has no KEGG from either tool nor a functional COG from EggNOG.

Table 16. Truth table manually generated using Microsoft Excel, logic of truth table used to develop the confidence rank system. An additional condition in the confidence system was required, this was not possible to illustrate in Table 15.



KEGG match	EggNOG KEGG	EggNOG COG	Kofam KEGG	Confidence value
y	y	y	y	1
y	y	n	y	1
n	y	y	y	2
n	y	n	y	2
	y	y	n	3
	n	y	y	3
	y	n	n	4
	n	n	y	4
	n	y	n	5
	n	n	n	0

The confidence ranking system was devised to determine whether the protein predictions of both the tools (EggNOG and KofamScan) were precise. The confidence value is based on 3 parameters- EggNOG providing a KEGG KO, EggNOG providing a functional COG and finally KofamScan providing a KEGG KO. Table 17 was generated to be inputted into the confidence system.

Table 17. Input file for Confidence ranking system (from Methanothrix soehngenii MAG). Output files from EggNOG and KofamScan (tsv) were merged using an outer join. The entire output file can be found in the Github repository, MRes-pipeline (https://github.com/louisegrimble/MRes-pipeline).

#query	seed_ortholog	evalue	score
JKJFLGDP_00001	1173027.Mic7113_1202	3.21E-85	275
JKJFLGDP_00002	756067.MicvaDRAFT_1231	4.63E-22	95.5
JKJFLGDP_00003	192952.MM_1348	0.000962	42.7
JKJFLGDP_00004	521011.Mpal_0288	3.16E-11	63.9
JKJFLGDP_00005	224325.AF_0894	2.55E-37	141
JKJFLGDP_00006	1094980.Mpsy_1669	1.30E-48	170
JKJFLGDP_00007	679926.Mpet_0612	3.29E-09	58.2
JKJFLGDP_00008	269797.Mbar_A1008	1.24E-14	75.5
JKJFLGDP_00009	1094980.Mpsy_1047	1.34E-26	105
JKJFLGDP_00010	1120936.KB907210_gene5880	1.00E-31	119

eggNOG_OGs
COG1506@1|root,COG1506@2|Bacteria,1G200@1117|Cyanobacteria,1H8T3@1150|Oscillatoriales
COG1506@1|root,COG1506@2|Bacteria,1G200@1117|Cyanobacteria,1H8T3@1150|Oscillatoriales
COG0018@1|root,arCOG00487@2157|Archaea,2XTWP@28890|Euryarchaeota,2N9C5@224756|Methanomicrobia
COG0018@1|root,arCOG00487@2157|Archaea,2XTWP@28890|Euryarchaeota,2N9C5@224756|Methanomicrobia
COG0018@1|root,arCOG00487@2157|Archaea,2XTWP@28890|Euryarchaeota,245YF@183980|Archaeoglobi
COG0018@1|root,arCOG00487@2157|Archaea,2XTWP@28890|Euryarchaeota,2N9C5@224756|Methanomicrobia
COG0018@1|root,arCOG00487@2157|Archaea,2XTWP@28890|Euryarchaeota,2N9C5@224756|Methanomicrobia
COG0018@1|root,arCOG00487@2157|Archaea,2XTWP@28890|Euryarchaeota,2N9C5@224756|Methanomicrobia
COG1842@1|root,arCOG04782@2157|Archaea,2XU11@28890|Euryarchaeota,2N9P1@224756|Methanomicrobia
COG1842@1|root,COG1842@2|Bacteria,2GP0K@201174|Actinobacteria,4EGYB@85012|Streptosporangiales

max_annot_lvl	COG_category	Description
1117|Cyanobacteria	E	PFAM Prolyl oligopeptidase family
1117|Cyanobacteria	E	PFAM Prolyl oligopeptidase family
224756|Methanomicrobia	J	Belongs to the class-I aminoacyl-tRNA synthetase family
224756|Methanomicrobia	J	Belongs to the class-I aminoacyl-tRNA synthetase family
183980|Archaeoglobi	J	Belongs to the class-I aminoacyl-tRNA synthetase family
224756|Methanomicrobia	J	Belongs to the class-I aminoacyl-tRNA synthetase family
224756|Methanomicrobia	J	Belongs to the class-I aminoacyl-tRNA synthetase family
224756|Methanomicrobia	J	Belongs to the class-I aminoacyl-tRNA synthetase family
224756|Methanomicrobia	K	PspA/IM30 family
201174|Actinobacteria	KT	PspA/IM30 family

Preferred_name	GOs	EC	KEGG_ko	KEGG_Pathway	KEGG_Module
dap2	-	-	-	-	-
dap2	-	-	-	-	-
argS	-	6.1.1.19	ko:K01887	ko00970,map00970	M00359,M00360
argS	-	6.1.1.19	ko:K01887	ko00970,map00970	M00359,M00360
argS	-	6.1.1.19	ko:K01887	ko00970,map00970	M00359,M00360
argS	-	6.1.1.19	ko:K01887	ko00970,map00970	M00359,M00360
argS	-	6.1.1.19	ko:K01887	ko00970,map00970	M00359,M00360
argS	-	6.1.1.19	ko:K01887	ko00970,map00970	M00359,M00360
-	-	-	ko:K03969	-	-
-	-	-	ko:K03969	-	-

KEGG_Reaction	KEGG_rclass	BRITE
-	-	-
-	-	-
R03646	RC00055,RC00523	ko00000,ko00001,ko00002,ko01000,ko01007,ko03016,ko03029
R03646	RC00055,RC00523	ko00000,ko00001,ko00002,ko01000,ko01007,ko03016,ko03029
R03646	RC00055,RC00523	ko00000,ko00001,ko00002,ko01000,ko01007,ko03016,ko03029
R03646	RC00055,RC00523	ko00000,ko00001,ko00002,ko01000,ko01007,ko03016,ko03029
R03646	RC00055,RC00523	ko00000,ko00001,ko00002,ko01000,ko01007,ko03016,ko03029
R03646	RC00055,RC00523	ko00000,ko00001,ko00002,ko01000,ko01007,ko03016,ko03029
-	-	ko00000
-	-	ko00000

KEGG_TC	CAZy	BiGG_Reaction	PFAMs
-	-	-	PD40,Peptidase_S9
-	-	-	PD40,Peptidase_S9
-	-	-	Arg_tRNA_synt_N,DALR_1,tRNA-synt_1d
-	-	-	Arg_tRNA_synt_N,DALR_1,tRNA-synt_1d
-	-	-	Arg_tRNA_synt_N,DALR_1,tRNA-synt_1d
-	-	-	Arg_tRNA_synt_N,DALR_1,tRNA-synt_1d
-	-	-	Arg_tRNA_synt_N,DALR_1,tRNA-synt_1d
-	-	-	Arg_tRNA_synt_N,DALR_1,tRNA-synt_1d
-	-	-	PspA_IM30
-	-	-	PspA_IM30

gene name	KO	thrshld	score	E-value	KO definition
---------	------	-------	------	---------	-------------
JKJFLGDP_00001	K03641	120.57	39.8	5.40E-11	TolB protein
JKJFLGDP_00001	K08676	573.57	23	1.90E-06	tricorn protease [EC:3.4.21.-]
JKJFLGDP_00001	K19732	171.77	11	0.022	LuxR family transcriptional regulator, activator of conjugal transfer of Ti plasmids
JKJFLGDP_00003	K13077	762.57	10.5	0.027	flavone synthase I [EC:1.14.20.5]
JKJFLGDP_00004	K01887	127.03	41.8	8.10E-12	arginyl-tRNA synthetase [EC:6.1.1.19]
JKJFLGDP_00004	K01886	508.87	14.8	0.00089	glutaminyl-tRNA synthetase [EC:6.1.1.18]
JKJFLGDP_00005	K01887	127.03	96.2	3.00E-28	arginyl-tRNA synthetase [EC:6.1.1.19]
JKJFLGDP_00006	K01887	127.03	81.5	8.30E-24	arginyl-tRNA synthetase [EC:6.1.1.19]
JKJFLGDP_00006	K04566	231.2	17.8	9.30E-05	lysyl-tRNA synthetase, class I [EC:6.1.1.6]
JKJFLGDP_00007	K01887	127.03	18.4	9.20E-05	arginyl-tRNA synthetase [EC:6.1.1.19]

4.12 Confidence ranking system output for Methanothrix soehngenii

The output from the confidence system is almost identical to Table 17, except with an additional column, ‘confidence_value’ (Table 18). This additional column consists of the confidence value that was assigned to each ORF based on the logic in the truth tables (Table 16).

Table 18. Output file generated by the confidence system (from Methanothrix soehngenii MAG) illustrates the confidence score of each ORF. The entire output file can be found in the Github repository, MRes-pipeline (https://github.com/louisegrimble/MRes-pipeline).

#query	seed_ortholog	evalue	score
JKJFLGDP_00001	1173027.Mic7113_1202	3.21E-85	275
JKJFLGDP_00002	756067.MicvaDRAFT_1231	4.63E-22	95.5
JKJFLGDP_00003	192952.MM_1348	0.000962	42.7
JKJFLGDP_00004	521011.Mpal_0288	3.16E-11	63.9
JKJFLGDP_00005	224325.AF_0894	2.55E-37	141
JKJFLGDP_00006	1094980.Mpsy_1669	1.30E-48	170
JKJFLGDP_00007	679926.Mpet_0612	3.29E-09	58.2
JKJFLGDP_00008	269797.Mbar_A1008	1.24E-14	75.5
JKJFLGDP_00009	1094980.Mpsy_1047	1.34E-26	105
JKJFLGDP_00010	1120936.KB907210_gene5880	1.00E-31	119

eggNOG_OGs
COG1506@1|root,COG1506@2|Bacteria,1G200@1117|Cyanobacteria,1H8T3@1150|Oscillatoriales
COG1506@1|root,COG1506@2|Bacteria,1G200@1117|Cyanobacteria,1H8T3@1150|Oscillatoriales
COG0018@1|root,arCOG00487@2157|Archaea,2XTWP@28890|Euryarchaeota,2N9C5@224756|Methanomicrobia
COG0018@1|root,arCOG00487@2157|Archaea,2XTWP@28890|Euryarchaeota,2N9C5@224756|Methanomicrobia
COG0018@1|root,arCOG00487@2157|Archaea,2XTWP@28890|Euryarchaeota,245YF@183980|Archaeoglobi
COG0018@1|root,arCOG00487@2157|Archaea,2XTWP@28890|Euryarchaeota,2N9C5@224756|Methanomicrobia
COG0018@1|root,arCOG00487@2157|Archaea,2XTWP@28890|Euryarchaeota,2N9C5@224756|Methanomicrobia
COG0018@1|root,arCOG00487@2157|Archaea,2XTWP@28890|Euryarchaeota,2N9C5@224756|Methanomicrobia
COG1842@1|root,arCOG04782@2157|Archaea,2XU11@28890|Euryarchaeota,2N9P1@224756|Methanomicrobia
COG1842@1|root,COG1842@2|Bacteria,2GP0K@201174|Actinobacteria,4EGYB@85012|Streptosporangiales

max_annot_lvl	COG_category	Description
1117|Cyanobacteria	E	PFAM Prolyl oligopeptidase family
1117|Cyanobacteria	E	PFAM Prolyl oligopeptidase family
224756|Methanomicrobia	J	Belongs to the class-I aminoacyl-tRNA synthetase family
224756|Methanomicrobia	J	Belongs to the class-I aminoacyl-tRNA synthetase family
183980|Archaeoglobi	J	Belongs to the class-I aminoacyl-tRNA synthetase family
224756|Methanomicrobia	J	Belongs to the class-I aminoacyl-tRNA synthetase family
224756|Methanomicrobia	J	Belongs to the class-I aminoacyl-tRNA synthetase family
224756|Methanomicrobia	J	Belongs to the class-I aminoacyl-tRNA synthetase family
224756|Methanomicrobia	K	PspA/IM30 family
201174|Actinobacteria	KT	PspA/IM30 family

Preferred_name	GOs	EC	KEGG_ko	KEGG_Pathway	KEGG_Module
dap2	-	-	-	-	-
dap2	-	-	-	-	-
argS	-	6.1.1.19	ko:K01887	ko00970,map00970	M00359,M00360
argS	-	6.1.1.19	ko:K01887	ko00970,map00970	M00359,M00360
argS	-	6.1.1.19	ko:K01887	ko00970,map00970	M00359,M00360
argS	-	6.1.1.19	ko:K01887	ko00970,map00970	M00359,M00360
argS	-	6.1.1.19	ko:K01887	ko00970,map00970	M00359,M00360
argS	-	6.1.1.19	ko:K01887	ko00970,map00970	M00359,M00360
-	-	-	ko:K03969	-	-
-	-	-	ko:K03969	-	-

KEGG_Reaction	KEGG_rclass	BRITE
-	-	-
-	-	-
R03646	RC00055,RC00523	ko00000,ko00001,ko00002,ko01000,ko01007,ko03016,ko03029
R03646	RC00055,RC00523	ko00000,ko00001,ko00002,ko01000,ko01007,ko03016,ko03029
R03646	RC00055,RC00523	ko00000,ko00001,ko00002,ko01000,ko01007,ko03016,ko03029
R03646	RC00055,RC00523	ko00000,ko00001,ko00002,ko01000,ko01007,ko03016,ko03029
R03646	RC00055,RC00523	ko00000,ko00001,ko00002,ko01000,ko01007,ko03016,ko03029
R03646	RC00055,RC00523	ko00000,ko00001,ko00002,ko01000,ko01007,ko03016,ko03029
-	-	ko00000
-	-	ko00000

KEGG_TC	CAZy	BiGG_Reaction	PFAMs
-	-	-	PD40,Peptidase_S9
-	-	-	PD40,Peptidase_S9
-	-	-	Arg_tRNA_synt_N,DALR_1,tRNA-synt_1d
-	-	-	Arg_tRNA_synt_N,DALR_1,tRNA-synt_1d
-	-	-	Arg_tRNA_synt_N,DALR_1,tRNA-synt_1d
-	-	-	Arg_tRNA_synt_N,DALR_1,tRNA-synt_1d
-	-	-	Arg_tRNA_synt_N,DALR_1,tRNA-synt_1d
-	-	-	Arg_tRNA_synt_N,DALR_1,tRNA-synt_1d
-	-	-	PspA_IM30
-	-	-	PspA_IM30

gene name	KO	thrshld	score	E-value
---------	------	-------	------	---------
JKJFLGDP_00001	K03641	120.57	39.8	5.40E-11
JKJFLGDP_00001	K08676	573.57	23	1.90E-06
JKJFLGDP_00001	K19732	171.77	11	0.022
JKJFLGDP_00003	K13077	762.57	10.5	0.027
JKJFLGDP_00004	K01887	127.03	41.8	8.10E-12
JKJFLGDP_00004	K01886	508.87	14.8	0.00089
JKJFLGDP_00005	K01887	127.03	96.2	3.00E-28
JKJFLGDP_00006	K01887	127.03	81.5	8.30E-24
JKJFLGDP_00006	K04566	231.2	17.8	9.30E-05
JKJFLGDP_00007	K01887	127.03	18.4	9.20E-05

KO definition	confidence_value
TolB protein	1
0	1
flavone synthase I [EC:1.14.20.5]	5
arginyl-tRNA synthetase [EC:6.1.1.19]	1
arginyl-tRNA synthetase [EC:6.1.1.19]	3
arginyl-tRNA synthetase [EC:6.1.1.19]	1
arginyl-tRNA synthetase [EC:6.1.1.19]	1
arginyl-tRNA synthetase [EC:6.1.1.19]	1
phage shock protein A 1 phage shock protein A	1

1037 ORFs have a confidence value of 1 (Table 19) in the *M. soehngenii* MAG. This accounts for 45.2% of the ORFs, meaning that both bioinformatic tools can provide an accurate annotation to approximately half of the ORFs in the *M. soehngenii* MAG. Considering that these results are from metagenomic data, a confidence of 45% is consistent with the current knowledge and tools. However, this highlights significant knowledge gaps in the subject area**.** Although the results generated by EggNOG and KofamScan are currently adequate, a higher percentage of ORFs being annotated accurately is preferable. A confidence value of 2 is also a satisfactory confidence prediction however, it is not entirely accurate. A much lower proportion (10.5%) of the ORFs scored a confidence value of 2. ORFs with a confidence value of 3 are not adequate, although they do provide some information about the general function of the ORF. ORFs with a confidence value of 4, 5, or 6, should be disregarded. These ORFs will undergo further annotation via an additional manual annotation (*Chapter 3.3.2*). To conclude, the ability of the pipeline to annotate and organise such large amounts of data is still beneficial.

Table 19. Summary of the confidence system analysis of each ORF in the Methanothrix soehngenii MAG.

Confidence Value	***Frequency***
1	1037
2	240
3	459
4	257
5	252
6	49

The output file from the confidence system is also illustrated in a bar plot (Figure 14), to visualise the distribution of confidence values across the entire *M. soehngenii* MAG.

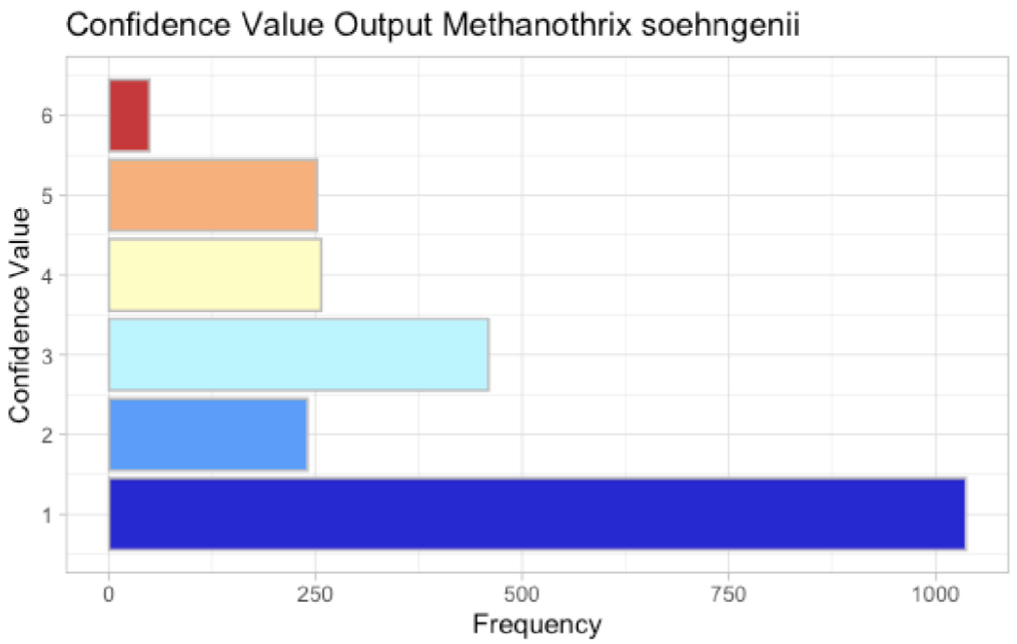


Figure 14. Barplot illustrating the confidence system output of the Methanothrix soehngenii MAG sequence. 1 = the highest confidence and 6= the lowest confidence. Colorblindness Guide R package was used to ensure the colours were colour blindness safe.

4.13 Manual annotation of the lowest quality ORFs of Methanothrix soehngenii

558 (23.4%) ORFs with the lowest confidence values (4, 5, or 6) were organised into a separate data frame (Table 20). These ORFs will undergo additional manual annotation in order to discover their function. After the ORFs have undergone this process, it is likely that more information will be discovered about the individual ORFs concerning their taxonomy, function, or sequence completeness.

Table 20. Output file generated by the confidence system (from Methanothrix soehngenii MAG) of each ORF with a low confidence score (i.e. confidence score of 4, 5 or 6). The entire output file can be found in the Github repository, MRes-pipeline (https://github.com/louisegrimble/MRes-pipeline).

query
JKJFLGDP_00002
JKJFLGDP_00011
JKJFLGDP_00072
JKJFLGDP_00079
JKJFLGDP_00083
JKJFLGDP_00084
JKJFLGDP_00085
JKJFLGDP_00086
JKJFLGDP_00087
JKJFLGDP_00093

4.14 Kraken2 output for low quality ORFs of Methanothrix soehngenii

Kraken2 output (Table 21) predicts the closest possible taxonomic annotation. This is illustrated in the first column, ‘U’ equates to unclassified, ‘K’ is kingdom, ‘P’ is phylum etc. A taxid is also provided to each sequence given, this will be ‘0’ if the protein is unclassified. Next is sequence length and the final column is the lowest common ancestor (LCA) mapping of each k-mer in the sequences. E.g. ‘562:13’ indicates that the first 13 kmers are mapped to the taxonomy ID 562. The majority of ORFs were predicted as an unclassified protein, meaning the confidence system was correct to define these as low quality open reading frames.

Table 21. Output file generated by Kraken2 for the Methanothrix soehngenii MAG. The entire output file can be found in the Github repository, MRes-pipeline (https://github.com/louisegrimble/MRes-pipeline).

Taxonomy	Sequence	Tax ID	sequence length	LCA
U	JKJFLGDP_00002	0	74	A:40
U	JKJFLGDP_00011	0	67	A:33
U	JKJFLGDP_00072	0	177	A:143
U	JKJFLGDP_00079	0	170	A:136
U	JKJFLGDP_00083	0	110	A:76
U	JKJFLGDP_00084	0	79	A:45
U	JKJFLGDP_00085	0	385	A:351
U	JKJFLGDP_00086	0	73	A:39
U	JKJFLGDP_00087	0	72	A:38
U	JKJFLGDP_00093	0	79	A:45
U	JKJFLGDP_00094	0	145	A:111

4.15 InterProScan output for low quality ORFs of Methanothrix soehngenii

InterProScan output file (Table 22) provides a functional annotation with a description of the function and a protein accession, which is a reference ID for the protein within the InterPro database. An E-value is also provided as well as a start and stop location of protein in the sequence. This would provide useful information for further analysis such as associative transcriptomics or alignment of query sequences against a known protein sequence. The type of analysis is also given in the output file. A Pfam analysis was performed on the ORFs in this MAG, which is the comparison of the sequence against the Pfam database and allows for the classification of protein domains and families (Finn, *et al.,* 2014).

Table 22. Output file generated by InterProScan for the Methanothrix soehngenii MAG. The entire output file can be found in the Github repository, MRes-pipeline (https://github.com/louisegrimble/MRes-pipeline).

JKJFLGDP_02396	8c0ef3f8ed19fe5467c97cb9a1a87a50	245	TIGRFAM
JKJFLGDP_02396	8c0ef3f8ed19fe5467c97cb9a1a87a50	245	Pfam
JKJFLGDP_00467	21ce23c07cd41d4f8eb6384d0ee85b60	195	Pfam
JKJFLGDP_00467	21ce23c07cd41d4f8eb6384d0ee85b60	195	Pfam
JKJFLGDP_00467	21ce23c07cd41d4f8eb6384d0ee85b60	195	PANTHER
JKJFLGDP_00444	54c411f7aed6f5bf4df7fcb2a17d3ca8	159	Pfam
JKJFLGDP_00444	54c411f7aed6f5bf4df7fcb2a17d3ca8	159	Gene3D
JKJFLGDP_00444	54c411f7aed6f5bf4df7fcb2a17d3ca8	159	TIGRFAM
JKJFLGDP_00444	54c411f7aed6f5bf4df7fcb2a17d3ca8	159	PANTHER
JKJFLGDP_00444	54c411f7aed6f5bf4df7fcb2a17d3ca8	159	CDD

TIGR02391	hypoth_ymh: TIGR02391 family protein	118	238	2.80E-29
PF09509	Protein of unknown function (Hypoth_ymh)	115	235	1.10E-31
PF01169	Uncharacterized protein family UPF0016	15	87	4.60E-17
PF01169	Uncharacterized protein family UPF0016	112	185	1.80E-22
PTHR12608	TRANSMEMBRANE PROTEIN HTP-1 RELATED	14	188	1.30E-23
PF04608	Phosphatidylglycerophosphatase A	51	147	4.60E-09
G3DSA:1.10.3760.10	-	-	158	2.80E-39
TIGR03161	ribazole_CobZ: alpha-ribazole phosphatase CobZ	7	148	2.70E-47
PTHR35336	ADENOSYLCOBINAMIDE AMIDOHYDROLASE	4	130	7.20E-11
cd06971	PgpA	6	148	1.07E-13

T	30/01/2023	IPR012654	Conserved hypothetical protein CHP02391	-
T	30/01/2023	IPR012654	Conserved hypothetical protein CHP02391	-
T	30/01/2023	IPR001727	Gdt1 family	-
T	30/01/2023	IPR001727	Gdt1 family	-
T	30/01/2023	IPR001727	Gdt1 family	-
T	30/01/2023	IPR007686	YutG/PgpA domain	GO:0006629|GO:0008962
T	30/01/2023	-	-	-
T	30/01/2023	IPR017577	Alpha-ribazole phosphatase, CobZ	-
T	30/01/2023	-	-	
T	30/01/2023	IPR007686	YutG/PgpA domain	GO:0006629|GO:0008962

4.16 Blast+ output for low quality ORFs of Methanothrix soehngenii

Blast+ output file generates a sequence ID of reference sequences, to identify the protein prediction within the NCBI database. An E-value is given, as well as the percentage of identical matches (pident), alignment length, stop and start of alignment of query sequence. Further analysis of the matches made by Blast+ can be made by the user. Due to technical difficulties with the Viking cluster, the Blast+ output files were not generated.

4.17 SeqKit output for low quality ORFs of Methanothrix soehngenii

A text file of the low quality ORFs that were predicted as a ‘hypothetical’ protein by PROKKA was generated (Table 23), using the Unix command ‘grep’. 302 (54.1%) low quality ORFs were predicted as hypothetical proteins by PROKKA.

Table 23. Output file generated by Snakemake rule ‘hypo_list’ (from Methanothrix soehngenii MAG), consists of all low quality ORFs that are annotated as ‘hypothetical’ proteins in the PROKKA annotation. The entire output file can be found in the Github repository, MRes-pipeline (https://github.com/louisegrimble/MRes-pipeline).

Query	Function
>JKJFLGDP_00002	hypothetical protein
>JKJFLGDP_00011	hypothetical protein
>JKJFLGDP_00072	hypothetical protein
>JKJFLGDP_00079	hypothetical protein
>JKJFLGDP_00083	hypothetical protein
>JKJFLGDP_00084	hypothetical protein
>JKJFLGDP_00086	hypothetical protein
>JKJFLGDP_00087	hypothetical protein
>JKJFLGDP_00093	hypothetical protein
>JKJFLGDP_00094	hypothetical protein

The final output generated by the pipeline is generated by SeqKit, which provides the sequence length of each of the low quality ORFs (Table 24). The sequence length of each ORF can provide an explanation for the lack of annotation provided by the numerous functional annotation tools. This could be due to incomplete proteins, which in metagenomic data could be caused by inaccurate shotgun sequencing or assembly of the MAGs (Bharti and Grimm, 2021). Another explanation for short sequence length could be naturally occurring missense mutations in the genome sequences, resulting in truncated protein sequences and are therefore unable to be identified.

Table 24. Output file generated by SeqKit tool, that includes the sequence length of each ORF that was defined as low quality by the confidence system in the Methanothrix soehgenii MAG.

Query	**Function**	**seq length**
JKJFLGDP_00002	hypothetical protein	74
JKJFLGDP_00011	hypothetical protein	67
JKJFLGDP_00072	hypothetical protein	177
JKJFLGDP_00079	hypothetical protein	170
JKJFLGDP_00083	hypothetical protein	110
JKJFLGDP_00084	hypothetical protein	79
JKJFLGDP_00085	ISNCY family transposase ISMac19	385
JKJFLGDP_00086	hypothetical protein	73
JKJFLGDP_00087	hypothetical protein	72
JKJFLGDP_00093	hypothetical protein	79

Chapter 5: Results (NCBI)

5.1 Summary of EggNOG and KofamScan annotations

The functional annotation generated by EggNOG and KofamScan were summarised respectively (Table 25). Within the NCBI dataset, EggNOG predicts a higher total number of proteins within each sequence. Often this is only by approximately 100 proteins, i.e. *S. aureus:* EggNOG 2435, KofamScan 2306 and *V. cholerae:* EggNOG 2681, KofamScan 2541. However, there are instances where there is a difference of 500 proteins i.e. *E. coli:* EggNOG 5121, KofamScan 4642. This is not indicative of the EggNOG annotation of the of M. soehngenii MAG (Table 13), which had a higher number of predictions from KofamScan rather than EggNOG. As mentioned in the MAG results (Table 13), the number of protein predictions is initially higher from KofamScan as three predictions are assigned to each ORF. The duplicates were removed to ensure reproducible results throughout the bioinformatic tools and confidence ranking system. The number of ORFs that were not assigned a KEGG KO number by EggNOG was fairly consistent throughout the NCBI dataset. A mean percentage of 34.7% of proteins in each genome sequence were not assigned a KEGG KO. This is indicative with the results of the *M. soehngenii* MAG (Table 13), where 31.5% of the proteins annotated by EggNOG were not assigned a KEGG KO. In contrast to EggNOG, KofamScan predicts a KEGG KO number for 100% of the ORFs within each genome sequence. KofamScan will give a prediction regardless of the quality of the prediction, therefore it is likely that the quality of the annotation by EggNOG is overall a higher quality than the KofamScan annotation.

Table 25. Summary of the EggNOG and KofamScan annotations (NCBI dataset).

	B. subtilis	**E. coli**	**P. aeruginosa**	**S. aureus**	**V. cholerae**
Proteins predicted EggNOG	3981	5121	6386	2435	2681
Proteins predicted KofamScan (incl. duplicates)	137536	141855	206627	76915	80230
Proteins predicted KofamScan	3758	4642	5932	2306	2541
Proteins without KEGG number (EggNOG)	1477 (37.1%)	1592 (31.1%)	2688 (42.1%)	855 (35.1%)	751 (28.0%)
Proteins without KEGG number (KofamScan)	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)

5.2 KEGG-Decoder output of NCBI dataset

The metabolic pathways present in all 5 genomes of the NCBI dataset is illustrated in a heatmap (Figure 15). *B. subtilis* and *S. aureus* genomes have a high number of genes related to the mixed acid: Lactate production, which is indicative of their known metabolism under anaerobic conditions. In the *E. coli* genome sequence, there was a high frequency of predicted genes involved in ethanol, acetate and formate production, which are common mixed acid fermentation products (Förster and Gescher, 2014). There were no genes involved in lactate production, this result is unexpected as E. coli does produce lactate via mixed acid fermentation (Förster and Gescher, 2014). It is likely that this error in annotation is due to incorrect sequencing of the *E. coli* sequence that is in the databases used by EggNOG and KofamScan. The heat map illustrated that *P. aeruginosa* metabolises via fermentation to produce only acetate and ethanol. There were no proteins present in the mixed acid: lactate pathway, even though *P. aeruginosa* can undergo pyruvate fermentation which results in the production of lactate (Kampers, *et al.,* 2021). This could indicate that *P. aeruginosa* is indirectly involved in producing lactate, rather than directly via mixed acid fermentation, resulting in a lack of lactate fermentation genes. Often, the annotation of certain pathways vary between KofamScan and EggNOG, for example, *S. aureus* undergoes mixed acid (acetate and ethanol) fermentation respectively. Indicative of literature (Fuchs, *et al.,* 2007), EggNOG illustrates the presence of genes within these two metabolic pathways, however, KofamScan does not. Not all genes in the mixed acid: acetate pathway are present in the KofamScan annotation, but all are present in the EggNOG annotation. This is likely due to the query sequences of these genes not aligning appropriately with the reference sequences in the databases. Efforts to ensure that small differences in sequences do not negatively impact the outcome of the annotation need to be implemented. None of the organisms within this dataset undergo methanogenesis, this is clearly illustrated in the heatmap, as there is a frequency of zero genes in each of the methanogenesis pathways. This confirms the annotations of EggNOG and KofamScan are consistent with the current bioinformatic tools and knowledge. All organisms in the NCBI dataset, except *S. aureus,* possess at least one flagella, this is also illustrated in the dendrogram. This result also ensures that the annotations predicted by EggNOG and KofamScan are accurate.

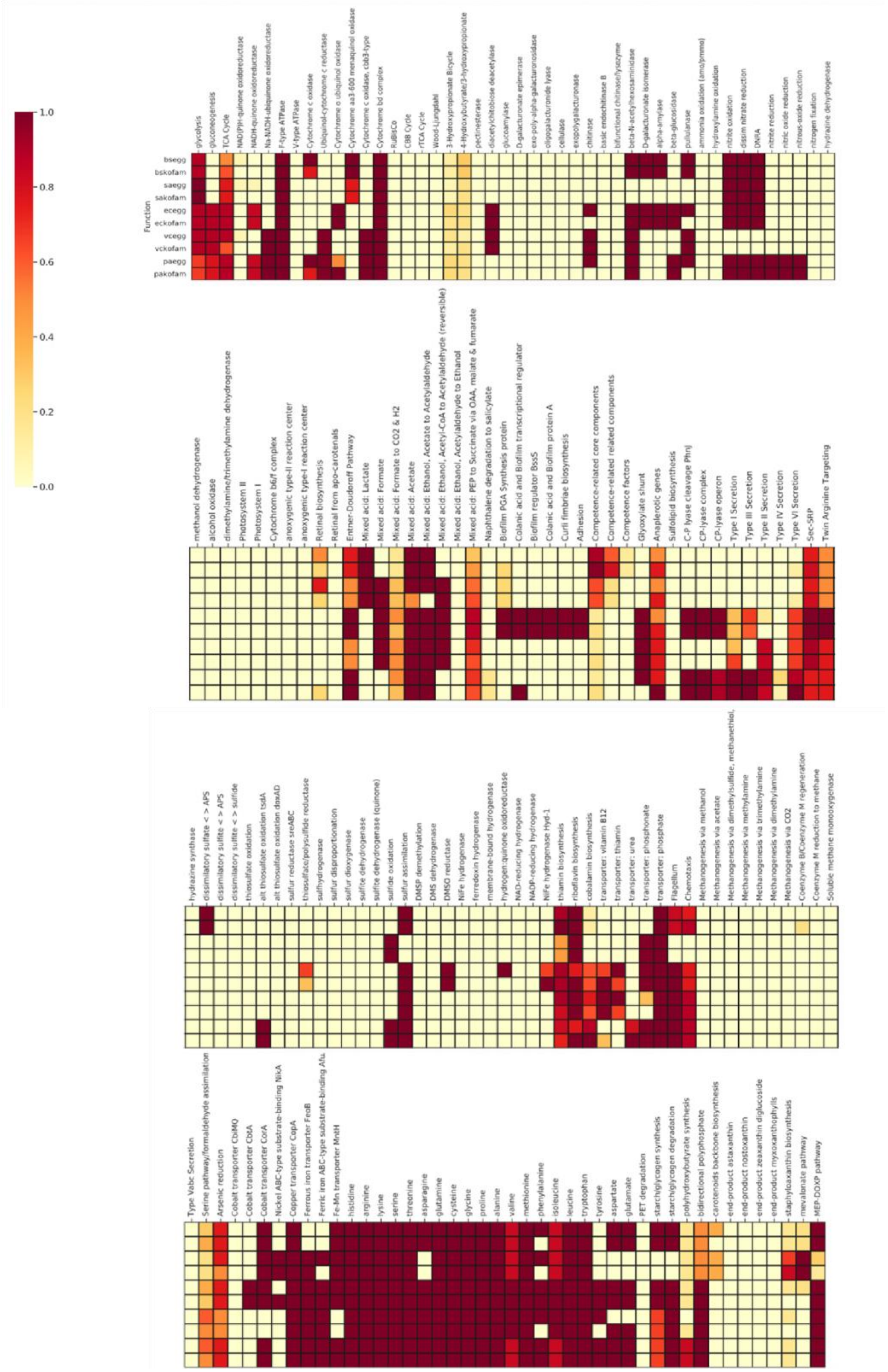


Figure 15. Heatmap generated by KEGG-Decoder illustrates the completeness of metabolic pathways in the NCBI dataset, recognised by KEGGmapper. Dark red - 1, white- 0, 1 is equivalent to complete pathway and 0 is equivalent to the entire pathway being absent. ec= E. coli, bs= B.subtilis, pa= P. aeruginosa, sa= S. aureus, vc= V. cholerae.

5.3 EggNOG E-value distribution of NCBI dataset

Unlike the results from the original MAG of *Methanothrix soehngenii* (Figure 12)*,* the E-value distribution of NCBI EggNOG predictions are not negatively skewed. However, all of the genomes in the NCBI dataset show a similar distribution (Figure 16), that displays no clear pattern or bias. The lack of pattern in the EggNOG output was not expected, as this group of genome sequences have a much higher percentage completion than the *M. soehngenii* MAG recovered from the AD site. Therefore, it would be expected that the predictions made by EggNOG would be more confident in the NCBI dataset. These genomes are of well characterised and highly researched organisms, and so the likelihood of the genes in these genomes to be present, is higher. However, this one result is not indicative of the entire functional analysis carried out in this pipeline.

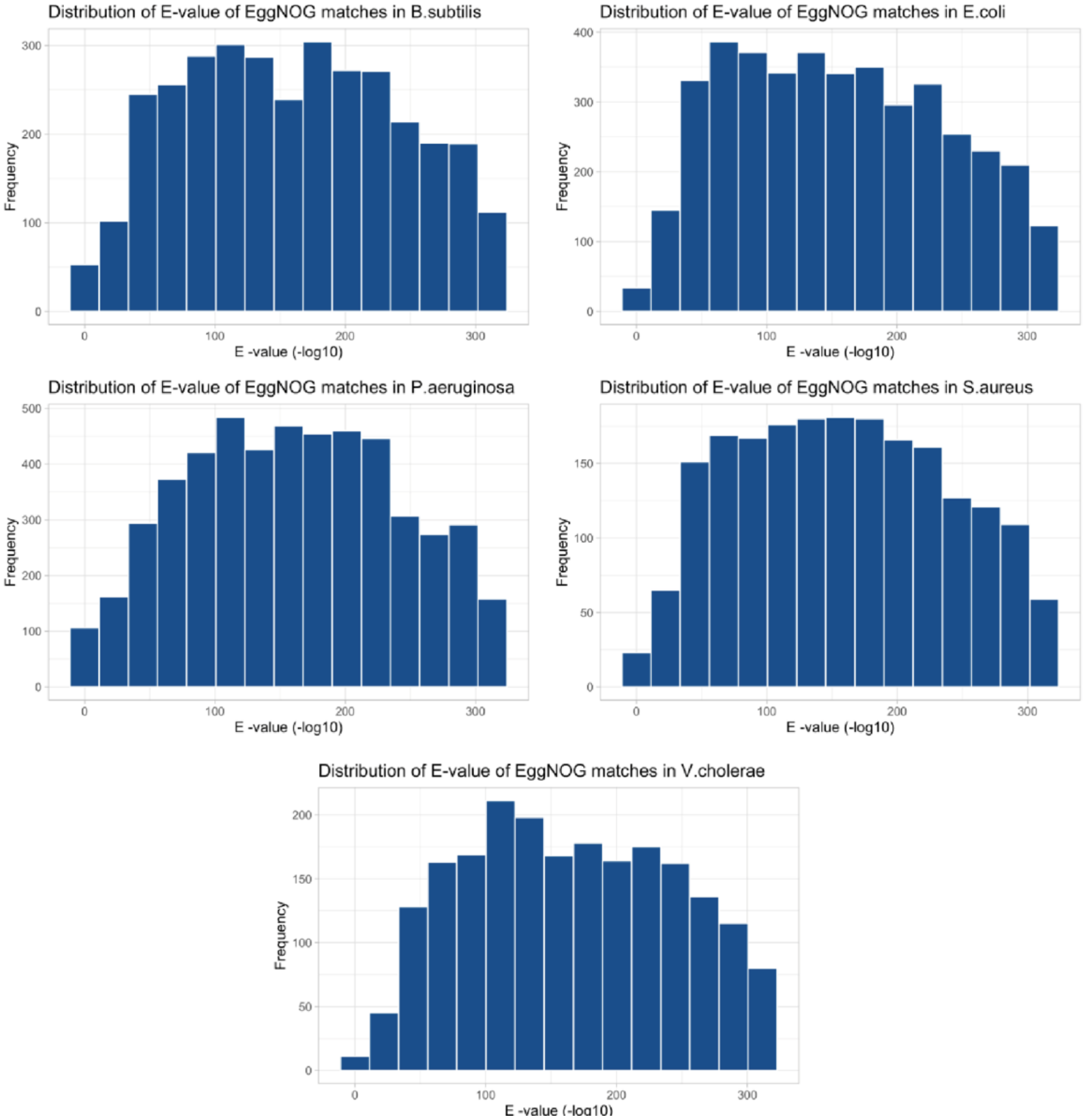


Figure 16. Histogram illustrating the distribution E-values of the annotations made by EggNOG of the NCBI dataset. Plots generated in R Studio using ggplot2, the data was logtransformed (-Log10). (a) Bacillus subtilis. (b) Escherichia coli. (c) Pseudomonas aeruginosa. (d) Staphylococcus aureus. (e) Vibrio cholerae.

5.4 EggNOG taxonomic analysis of Bacillus subtilis

The *B. subtilis* taxonomy plot (Figure 17) shows that the proteins present in this genome are derived from seven different taxonomic groups. Four of which provided a range of E-values indicating there are multiple proteins within these groups; *Firmicutes, Bacilli, Bacteria & Siphoviridae*. The other three taxonomic groups only contain a single E-value, suggesting a small number of proteins were predicted within the groups, *Thermotogae, Clostridia and Proteobacteria*. The taxonomy of *Bacillus* is as follows; Kingdom, *Bacteria*; Phylum, *Bacillota;* Class, *Bacilli;* Order, *Bacillales*; Family, *Bacillaceae*; Genus, *Bacillus*; Species, *Bacillus subtilis* group (Schoch, *et al.,* 2020). A paper by Huerta-Cepas, J. *et al.* states that "recent speciations lead to smaller and usually more functionally specific sets of orthologs". Therefore, proteins from taxonomic groups that are more closely related to the organism will most likely be unique to the organism of interest. Further analysis of genes within these taxonomic groups would help to identify genes of interest, i.e. E-value would determine the quality of the prediction made by EggNOG. *Bacilli* is the taxonomic group (Order) that is most closely related to *B. subtilis*, and would likely contain genes specific to its function compared to the other taxonomic groups.

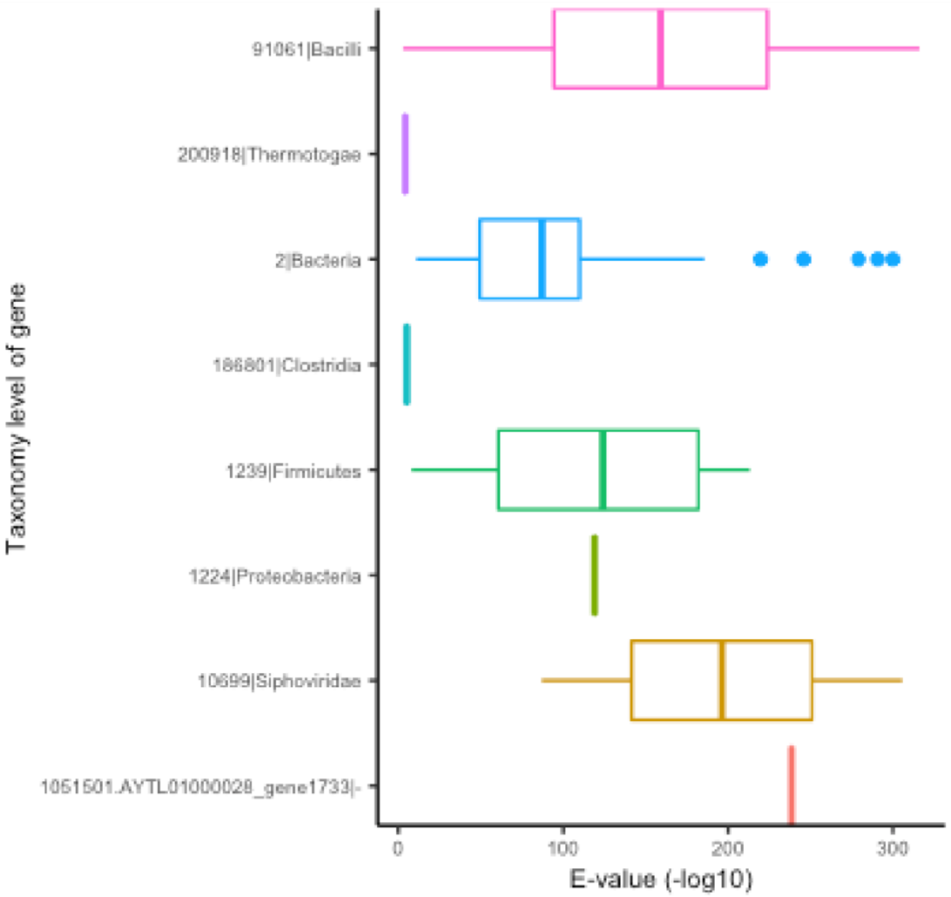


Figure 17. Boxplot illustrating the taxonomic analysis generated by EggNOG compared against the E-value of each ORF, for the Bacillus subtilis genome from NCBI. The plot was generated using ggplot2 in R studio and the data was log-transformed (-Log10).

5.5 EggNOG taxonomic analysis of Escherichia coli

EggNOG has predicted proteins from eight different taxonomic groups (Figure 18) in the *E. coli genome sequence*. These groups contain multiple E-values, suggesting there are multiple proteins within these groups. The taxonomic groups are *Caudovirales, Bacteria, Gammaproteobacteria, Proteobacteria, Podoviridae, Siphoviridae* & *Viruses*. Again, the ORFs from the closely related taxonomic groups are likely to contain genes of interest as they are most likely to be unique to the organism. 511145.b4204| is not a taxonomic level, this is seen again for multiple proteins in this figure- 155864.EDL933_2987|, 155864.EDL933_2515|, 155864.EDL933_2453|, 155864.EDL933_2442|, 155864.EDL933_2262|, 155864.EDL933_2013|, 155864.EDL933_1920, 155864.EDL933_1407, 155864.EDL933_1307, 155864.EDL933_1087, 155864.EDL933_0889. These are placed in the 'max_annot_lvl' column of the EggNOG output file (Table 10), although they should hypothetically be found in the preceding column, 'eggNOG_OGs. There is no taxonomy level provided to these particular ORFs, and upon further manual investigation via KEGGmapper, the description "duf" or "domain of unknown function" is typically given to these proteins. Therefore these should be discarded from the output results. There are several individual proteins that are classified as the following taxonomic groups *Metazoa, Betaproteobacteria, Alphaproteobacteria, Acidithiobacillales, Neisseriales*. The taxonomy of *E. coli* is as follows; Kingdom, *Bacteria*; Phylum, *Pseudomonadota*; Class, *Gammaproteobacteria*; Order, *Enterobacterales*; Family, *Enterobacteriaceae*; Genus, *Escherichia* (Schoch, *et al.,* 2020). *E. coli* belongs to the *Gammaproteobacteria* Order, and is the closest group related to *E. coli,* with multiple genes present in the EggNOG annotation. This group of proteins is most likely to have genes of interest of *E. coli* as they are most closely related.

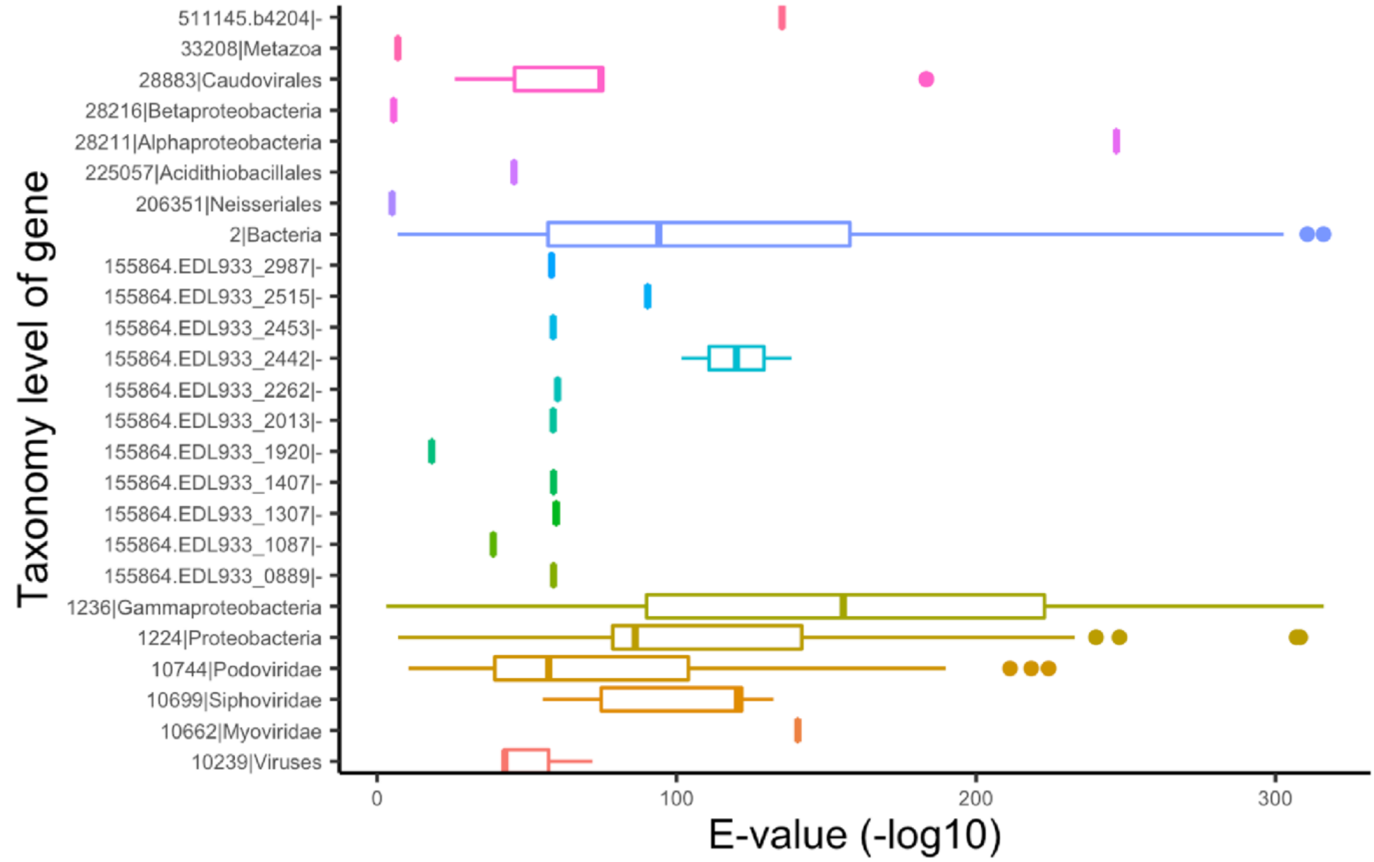


Figure 18. Boxplot illustrating the taxonomic analysis generated by EggNOG compared against the E-value of each ORF, for the Escherichia coli genome from NCBI. The plot was generated using ggplot2 in R studio and the data was log-transformed (-Log10).

5.6 EggNOG taxonomic analysis of Pseudomonas aeruginosa

There are four groups in this figure (Figure 19) that are not taxonomic groups, 287.DR97_1398|, 287.DR97_1396|, 287.DR97_1393|, 287.DR97_1031|. These groups are not defined by a taxonomic group and are most likely proteins with an unknown function. Again, this is due to an error with EggNOG output file formatting. There are then nine individual E-values, indicating a small number of proteins are predicted within these taxonomic groups. These groups are *Bacilli, Deltaproteobacteria, Neisseriales, Nitrosomonadales, Thermodesulfobacteria, Clostridia, Aeromonadales, Deinococcusthermus* and *Inoviridae*. Finally there are many taxonomic groups that contain multiple Evalues per taxonomic group, these groups include *Bacteroidetes, Caudovirales, Betaproteobacteria, Alphaproteobacteria, Rhodocyclales, Sphingomonadales, Rhodospirallales, Actinobacteria, Bacteria, Pasteurellales, Vibrionales, Oceanospirillales, Xanthomonadales, Chromatiales, Gammaproteobacteria, Proteobacteria, Cyanobacteria* & *myoviridae*. The taxonomy of *P. aeruginosa* is as follows; Kingdom, *Bacteria*; Phylum, *Pseudomonadota*; Class, *Gammaproteobacteria*; Order, *Pseudomonadales*; Family, *Pseudomonadaceae*; Genus, *Pseudomonas*; Species, *Pseudomonas aeruginosa* group (Schoch, *et al.,* 2020). The closest taxonomic group to *P. aeruginosa* is *Gammaproteobacteria*, the proteins from this group are likely to be specific to the function of this organism.

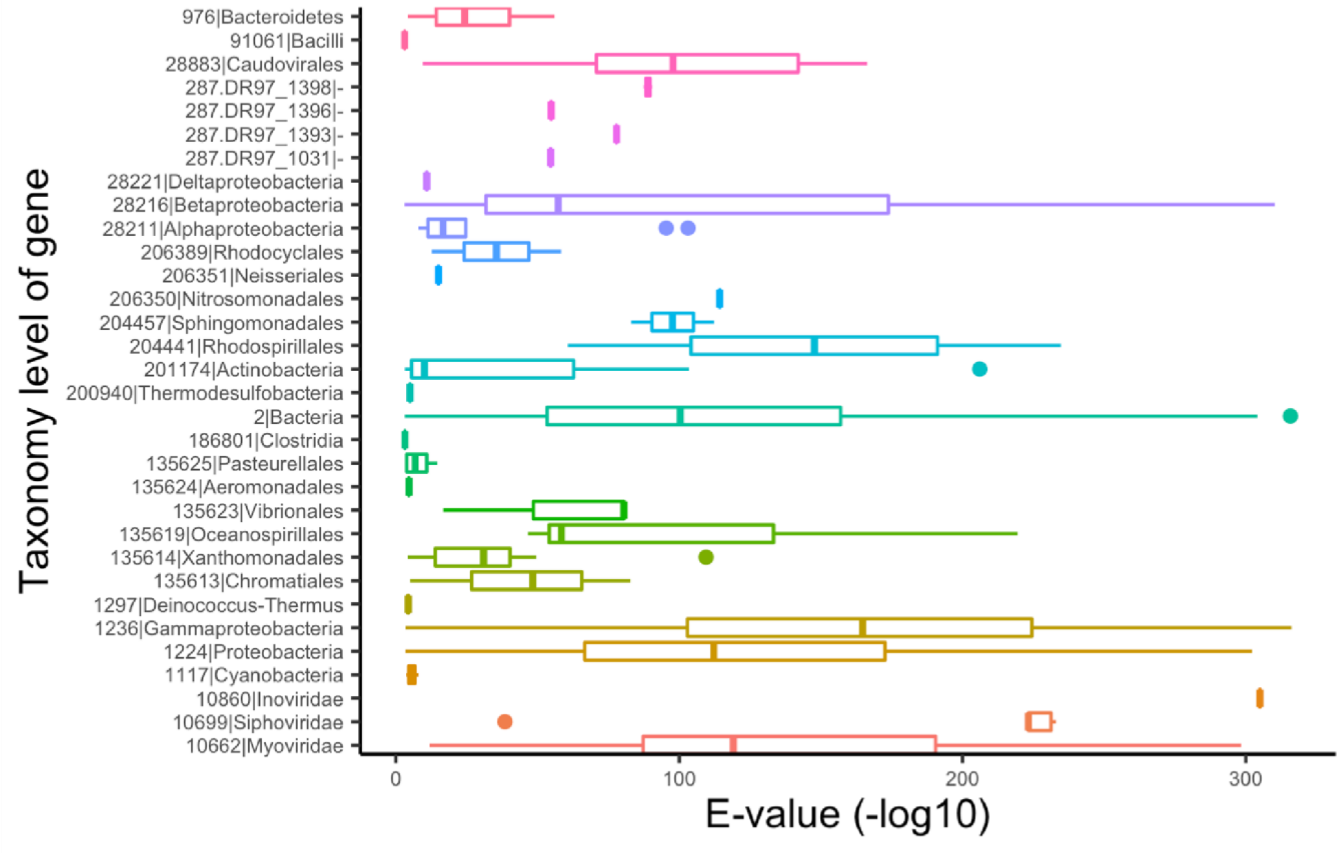


Figure 19. Boxplot illustrating the taxonomic analysis generated by EggNOG compared against the E-value of each ORF, for the Pseudomonas aeruginosa genome from NCBI. The plot was generated using ggplot2 in R studio and the data was log-transformed (Log10).

5.7 EggNOG taxonomic analysis of Staphylococcus aureus

The *S. aureus* EggNOG taxonomy plot (Figure 20) illustrates that there are five taxonomic groups, *Bacilli, Caudovirales, Bacteria, Firmicutes & Siphoviridae* that provided a range of Evalues. This indicates there are multiple proteins within these groups. Then, there are two groups, *Clostridia and viruses*, that both provide a single E-value each. Finally, there are five groups that are not taxonomic groups but in fact an error due to the EggNOG output file not being parsed in R correctly. These values should be disregarded, like previously mentioned for the other taxonomy plots. These are shown in the figure as 1280.SAXN108_2110|, 1280.SAXN108_0492|, 1280.SAXN108_0482, 1280.SAXN108_0481, 1280.SAXN108_0476. The taxonomy of *S. aureus* is as follows: Kingdom, *Bacteria*; Phylum, *Bacillota*; Class, *Bacilli*; Order, *Bacillales*; Family, *Staphylococcaceae*; Genus, *Staphylococcus* (Schoch, *et al.,* 2020). *Bacilli* is the taxonomic Class present from the EggNOG output that is most closely related to *S. aureus*. *Bacilli* is likely to contain genes of interest that are more targeted at the specific function of the organism.

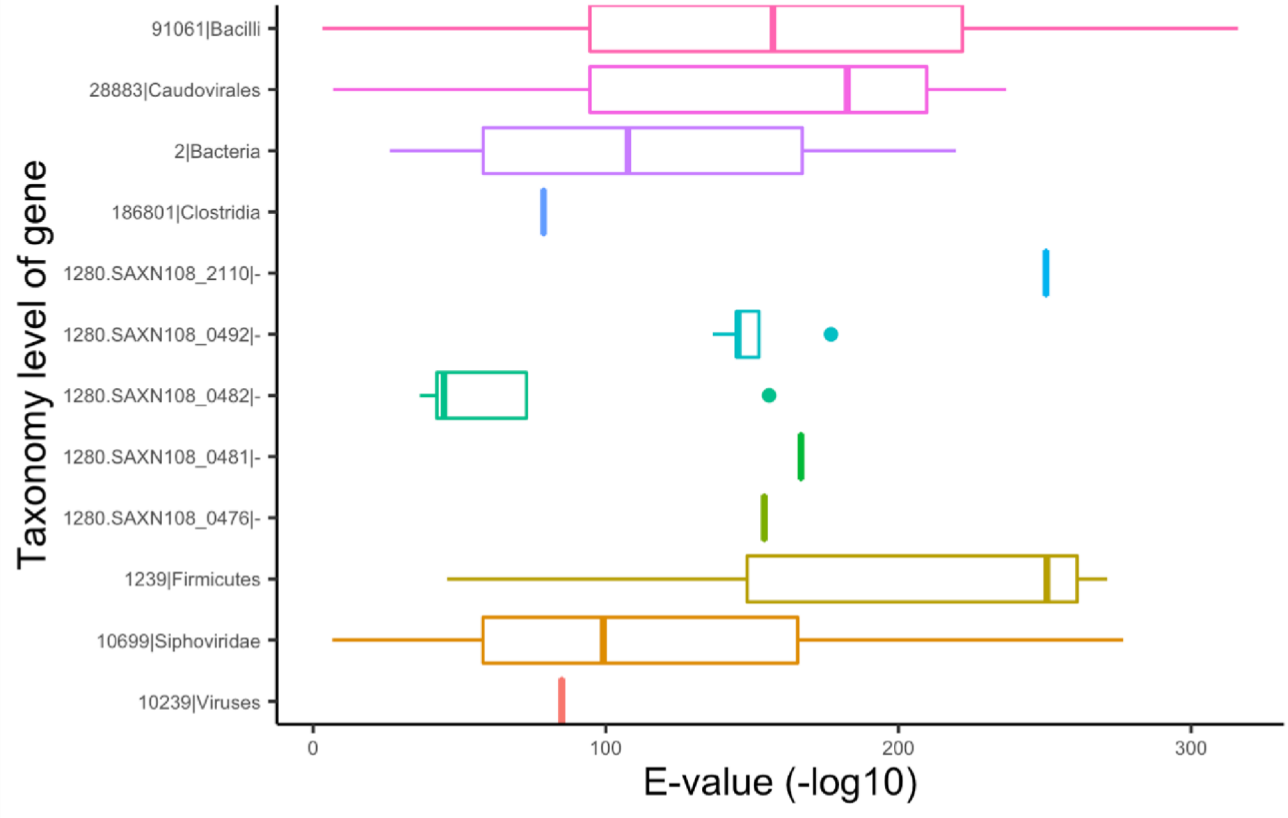


Figure 20. Boxplot illustrating the taxonomic analysis generated by EggNOG compared against the E-value of each ORF, for the Staphylococcus aureus genome from NCBI. The plot was generated using ggplot2 in R studio and the data was log-transformed (-Log10).

5.8 EggNOG taxonomic analysis of Vibrio cholerae

The *V. cholerae* taxonomy plot (Figure 21) has ten different taxonomic groups that provide a range of E-values, indicating there are multiple proteins within these groups. The groups include *Bacteroidetes, Bacilli, Epsilonproteobacteria, Betaproteobacteria, Alphaproteobacteria, Bacteria, Vibrionales, Oceanospirillales, Gammaproteobacteria & Cyanobacteria*. There are then, many individual E-values, and therefore proteins that have been classified within individual taxonomic groups. These groups include *Rhodocyclales, Sphingomonadales, Clostridia, Aeromonadales, Cyanobacteria, Inoviridae, Podoviridae, Siphoviridae & viruses*. The taxonomy of *V. cholerae* is as follows; Kingdom, *Bacteria*; Phylum, *Pseudomonadota*; Class, *Gammaproteobacteria*; Order, *Vibrionales*; Family, *Vibrionaceae*; Genus, *Vibrio* (Schoch, *et al.,* 2020). Therefore the taxonomic group *Vibrionales* (Order) is most likely to have genes of interest, as it is the group present that is most closely related to *V. cholerae*. The genes of interest are dependent on the users' intentions of the pipeline. In this context, the genes of interest are involved in anaerobic digestion and the related metabolic processes i.e. mixed acid fermentation pathways.

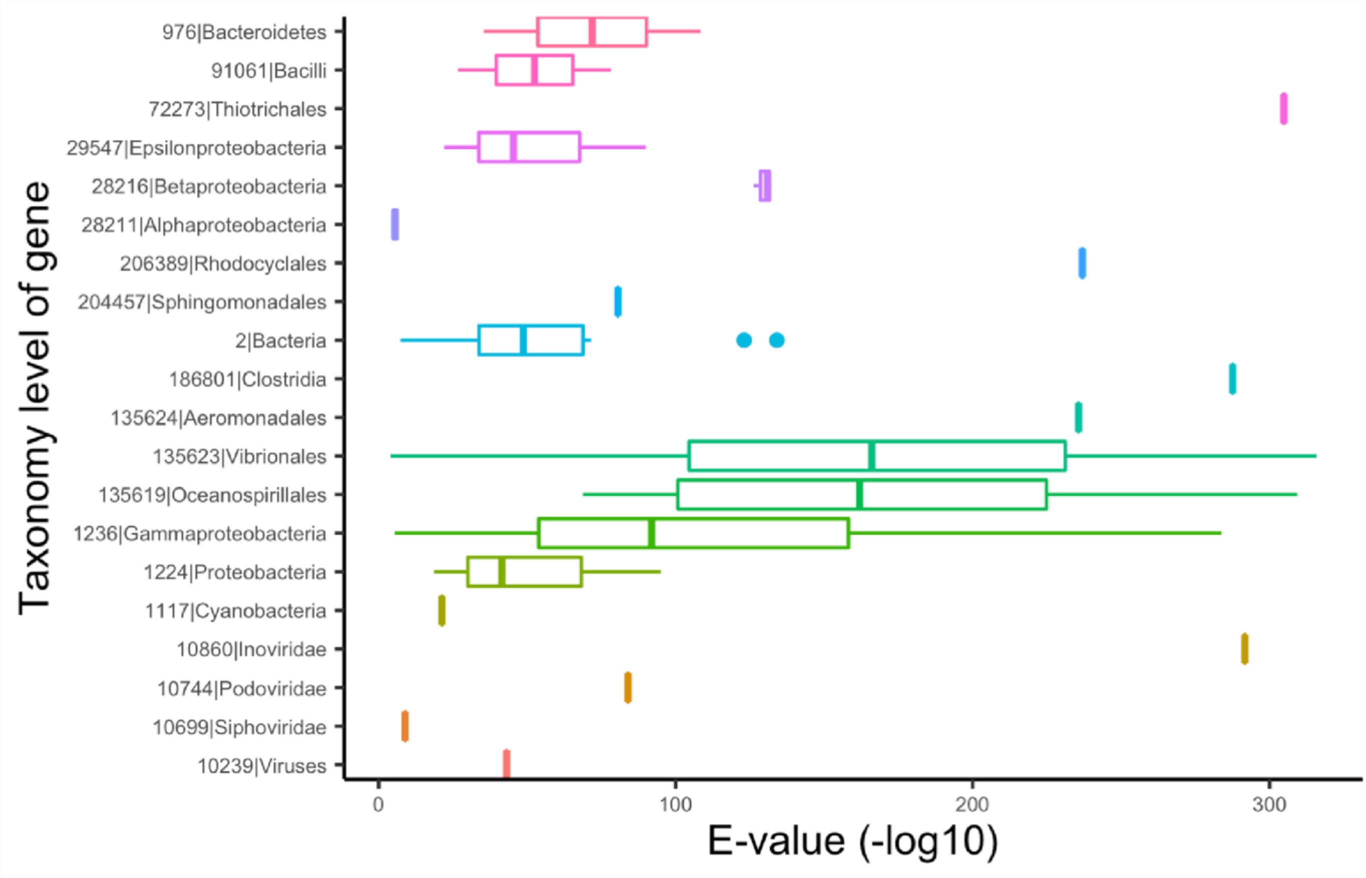


Figure 21. Boxplot illustrating the taxonomic analysis generated by EggNOG compared against the E-value of each ORF, for the Vibrio cholerae genome from NCBI. The plot was generated using ggplot2 in R studio and the data was log-transformed (-Log10).

5.9 The S functional COG is highly abundant within the NCBI dataset.

A trend seen in all of the plots (Figure 22), in the sequences in the NCBI dataset is a significantly high gene frequency in the S COG. All genome sequences exhibit very similar patterns of functional COGs considering their taxonomy varies. S functional COGs are proteins of unknown function (Chen, Xia and Li, 2022). Other common COGs in the NCBI dataset are J (Translation, ribosomal structure, and biogenesis) and K (transcription). However, these proteins are involved in basic cellular functions. Functional COGs of interest for this dataset, in relation to metabolism under anaerobic conditions within AD, would be C (energy production and conversion) and P (inorganic ion transport and metabolism).

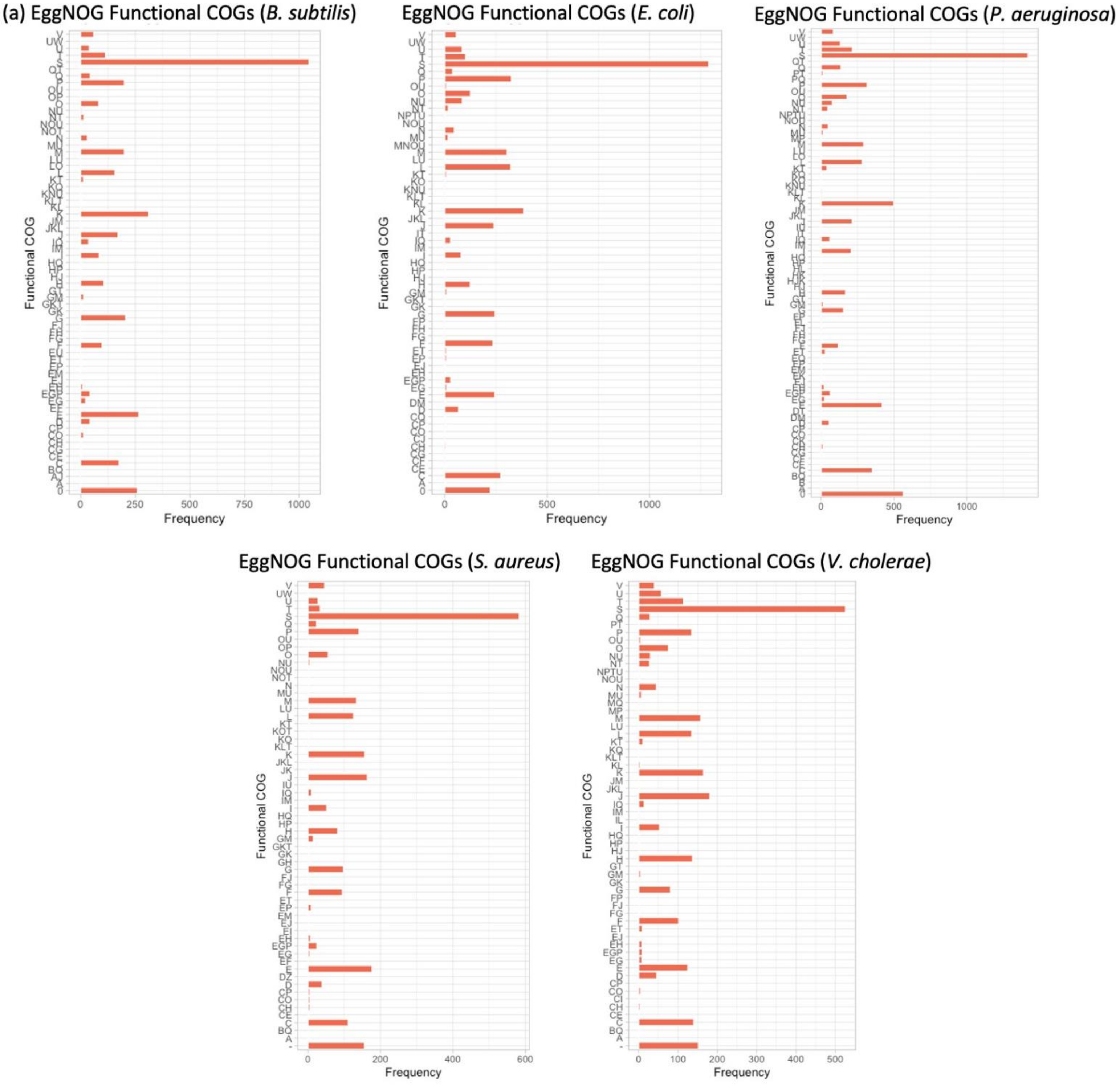


Figure 22. Bar plot illustrating the frequency of genes assigned to Functional COGs for the NCBI dataset from the EggNOG output data. Plot generated in R Studio using ggplot2. (a) EggNOG output for Bacillus subtilis. (b) EggNOG output for Escherichia coli. (c) EggNOG output for Pseudomonas aeruginosa. (d) EggNOG output for Staphylococcus aureus. (e) EggNOG output for Vibrio cholerae.

5.10 Confidence ranking system output for NCBI dataset.

The confidence ranking system, devised using Python, generated confidence values for all of the ORFs in the genome sequences of the NCBI dataset (Table 26). A mean percentage of 58.7% of genes were given a confidence score of 1, which is the most confident score. Meaning the functional prediction by both EggNOG and KofamScan is correct for approx. 60% of ORFs in these genomes. An average of 5.3% of ORFs across the NCBI dataset were given a confidence score of 2, which is the second highest confidence value of proteins within each sequence.

Table 26. Summary of the confidence system analysis of each ORF in the NCBI dataset.

Confidence Value	*Bacillus subtilis*	*Escherichia coli*	*Pseudomonas aeruginosa*	Staphylococcus aureus	*Vibrio cholerae*
1	2243	3295	3337	1404	1828
2	257	225	345	173	101
3	694	659	1296	396	373
4	254	154	437	134	108
5	541	726	848	309	237
6	96	115	294	56	73

The general trend seen across all of the genome sequences in the NCBI dataset (Figure 23) is that the majority of ORFs were scored a 1. Very few ORFs were given a score 2, however these ORFs are still satisfactory. An average of 19.8% of ORFs were given a low confidence value, i.e. 4, 5 or 6, in the NCBI dataset. ORFs with these confidence scores (4/5/6) do not have an adequate annotation from EggNOG or KofamScan, these annotations should be disregarded.

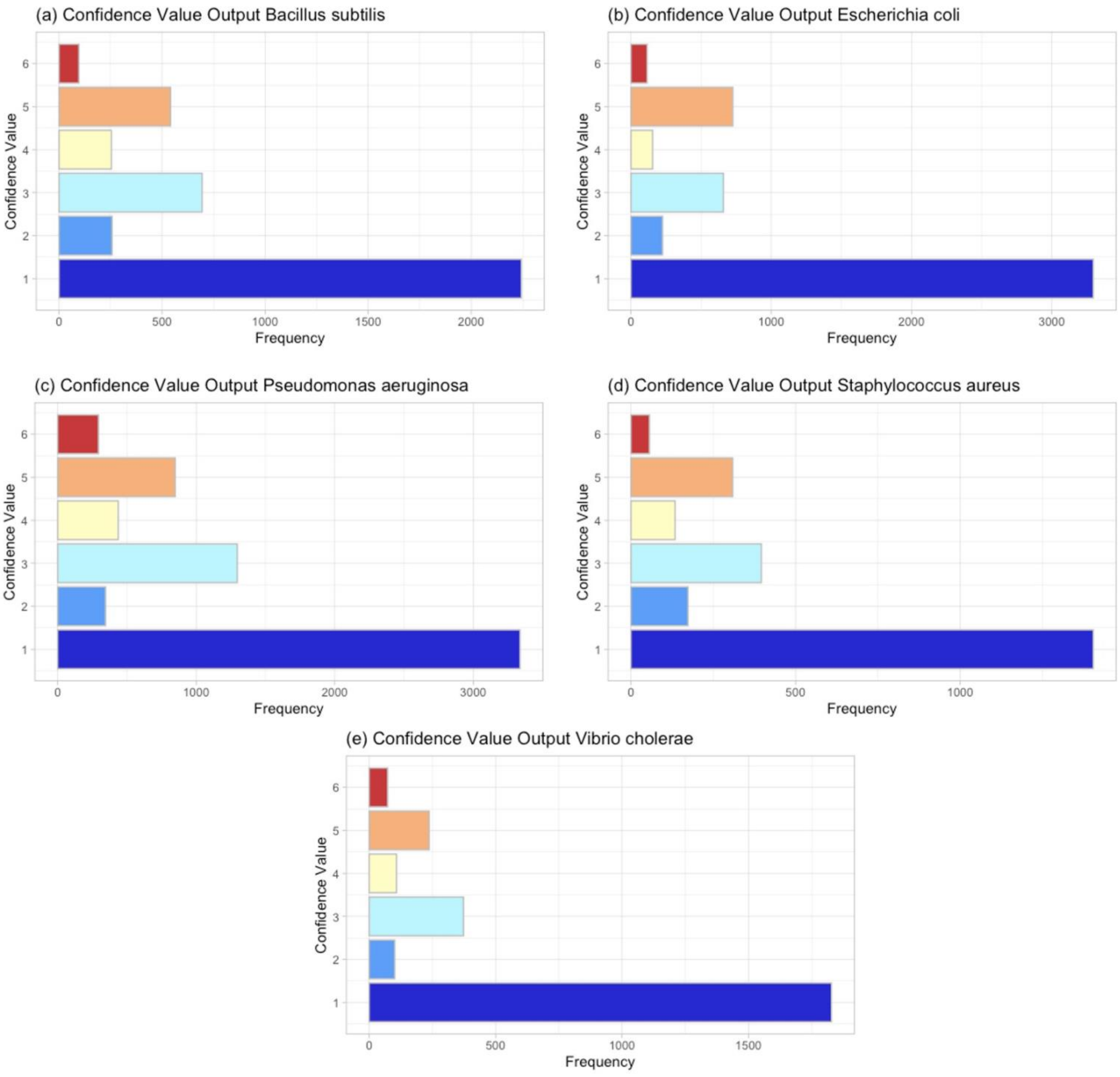


Figure 23. Bar plot illustrating the frequency of genes assigned to Confidence values based on the confidence ranking system that was devised on the NCBI dataset. 1 = the highest confidence and 6= the lowest confidence. Plots generated in R Studio using ggplot2. (a) Confidence value output for Bacillus subtilis. (b) Confidence value output Escherichia coli. (c) Confidence value output for Pseudomonas aeruginosa. (d) Confidence value output for Staphylococcus aureus. (e) Confidence value output for Vibrio cholerae.

5.11 Manual annotation of the lowest quality ORFs of NCBI dataset.

The low quality ORFs determined by the confidence system (Supplementary data S.2) are passed through additional functional annotation tools, InterProScan (Supplementary data

S.3) and taxonomic annotation tool Kraken2 (Supplementary data S.4). These output files can be found within the Supplementary data.

5.12 SeqKit outputs for low quality ORFs of NCBI dataset.

The number of hypothetical proteins from the annotation provided by PROKKA was achieved by using the grep command (Supplementary data S.5). *B. subtilis, E. coli, P. aeruginosa, S. aureus,* and *V. cholerae,* have *543, 746, 1147, 334, and 3317,* ORFs annotated as hypothetical proteins respectively.

The SeqKit output illustrates the sequence length (Supplementary data S.6) of all low quality ORFs. This data can be found within the Supplementary Data, and can provide an explanation for the lack of annotation provided by the bioinformatic tools in the pipeline.

Chapter 6: Accessibility

The inappropriate use of colours in data visualisation is a recurring often overlooked issue (Crameri, Shephard and Heron, 2020), not only for the average consumer of the data, but especially for people with colour blindness. The misuse of colour causes misinterpretation of data and therefore scientific findings (Crameri, Shephard and Heron, 2020). Adobe colour (Adobe, 2014) offers a web-based simulator of desired colours, showing the perception of certain colour palettes via different types of colour blindness (i.e. protanopia, tritanopia and deuteranopia). This ensures that there is a high contrast between colours.

ColorBlindness (Ou, 2021) is an R package that provides groups of suitable colours, for data visualisation, to aid users with colour blindness, by generating an example of the desired plot within the ranges of visibility of different types of colour blindness. This provides an easier solution for others developing data visualisation using R, rather than using a third party tool such as Adobe Color. There are other R packages that account for colour blindness when adding colour to figures generated in R Studio e.g. ColorBrewer (Neuwirth and Neuwirth, 2014) and viridis (Garnier, *et al.,* 2021). Upon use of this tool, the colour selected for the plots in this pipeline (Figure 24) were selected by using a colour palette provided by the ColorBlindness R package, that are appropriate for those with colour blindness. The taxonomy plots outputted by EggNOG, are not in a colour-blindness safe colour palette. This is due to the number of colours required to be defined, for the generation of this figure, differs each time the pipeline generates output files.

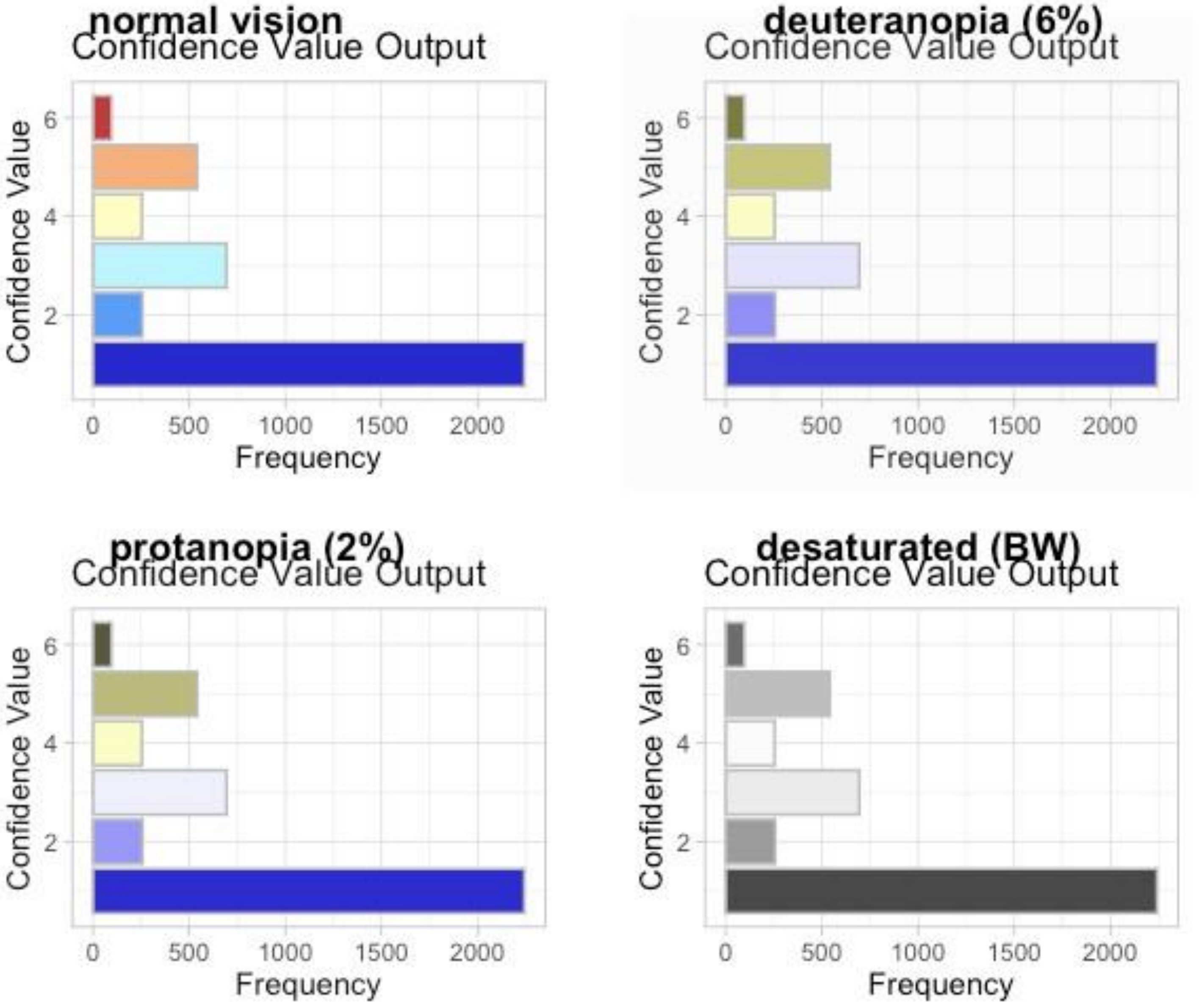


Figure 24: Schematic illustrating the use of the colorblindness guide R package, to ensure the colours utilised in the figure are safe for users with colour blindness.

The heatmap generated by KEGG-Decoder (Figure 9, Figure 15) has a colour palette that ranges from red to orange to yellow to white. Within the yellow to white region of the heatmap, it becomes increasingly difficult to determine whether there is any presence of proteins or whether all genes within this pathway are absent in this organism. The use of a colour palette that is accessible to a wider audience would improve the accessibility of this tool, and of this pipeline. Adding the option of a pattern alongside the colour would also improve the accessibility of this figure(Guha, Fertig and Deshpande, 2022).

Chapter 7: Discussion

7.1 Comparison of EggNOG and KofamScan functional annotation.

Regardless of which tool (EggNOG or KofamScan) predicts the most total proteins, it is the quality of the predictions made that makes it more valuable as a functional annotation tool. EggNOG does not give KEGG KO to each ORF, but it does mean that the ones that are predicted are definitely more likely to be correct.

There is a significant limitation with EggNOG outputs (Table 10), as ORFs are often given more than one KEGG KO. This is a result of the lack of a one-to-one relationship between protein function and KEGG KO. This is likely due to some proteins being ubiquitous and found within multiple biological pathways and providing different functions. Upon further investigation of the multiple KEGG KOs, they tend to be KOs of varying function. The issue is that for both KEGG-Decoder and the confidence ranking system, only one KEGG KO can be inputted per ORF. Making a decision on which KEGG KO remains, and which does not, was challenging. This was decided by the first KEGG KO within the annotation. As a result, critical information concerning the function of an organism can be lost. This is similarly true of the KofamScan output file that predicts three repeats of each ORF in the query sequence, where each repeat is given a different KEGG KO. The duplicates in the KofamScan output were removed in order to parse this output file into the confident ranking system

7.2 Comments on certain outputs and how they could be improved.

The EggNOG output (Figure 13) illustrates the distribution of the E-value of proteins within each functional COGs, however, these plots do not illustrate the exact frequency of proteins within each of the COGs. The addition of a plot of the frequency of proteins in each functional COG, or even a table including these frequencies would be beneficial, to show the frequency of proteins within COGs of interest. Targeting COGs of interest specifically within certain genomes, could be conducted manually, by defining areas of interest before utilising the pipeline. For example, in the context of microbial metabolism within AD. Functional COGs C, P and H are of interest as they are involved in metabolism pathways that are typically found within AD (Chen, Xia and Li, 2022).

Figure 17-20 illustrates the distribution of E-values within each taxonomic group of predicted proteins. However, a table summarising the frequency of proteins from each taxonomic group would help to target taxonomic groups of interest.

The heatmap, generated by KEGG-Decoder (Figure 9, Figure 15), does not quantify the frequency of proteins within each metabolic pathway, therefore it is not certain how many proteins are required for each metabolic pathway. Instead, an estimate on the presence of proteins is based on the completeness of each pathway as there cannot be an accurate comparison between the annotation of certain metabolic pathways, as the data is qualitative. A table generated to count the frequency of proteins that were predicted per pathway alongside the number of proteins found within each pathway, would provide quantitative data. There are instances (e.g. for the pathways regarding the amino acids such as phenylalanine and asparagine) where a complete annotation of a pathway is given by one tool and not another. This is likely due to the reference sequence of the genes within that database being lower in quality or dissimilar to the query sequence. To overcome this issue, multiple references of the same sequence should be inputted into the databases so the matching of the sequences is not as stringent and allows for slight changes in sequences. This is seen by both EggNOG and KofamScan, therefore it is not a concern for the quality of the annotations provided by the tool but the databases they refer to. This also highlights that the pipeline does not allow for contamination of MAGs, which could explain the lower number of high quality annotations of the MAG, in comparison to the NCBI dataset of monocultured strains.

7.3 Comparison of outputs between MAG data and NCBI dataset.

45.2% of proteins, in the MAG sequence (Table 19), were given a confidence score of 1 however, an average of 58.7% (Table 26) of proteins in the NCBI dataset were given a confidence score of 1. This result is expected as the NCBI dataset consists of high quality genomes that are more complete than the *M. soehngenii* MAG. The genomes in the NCBI dataset have also not undergone shotgun sequencing, therefore there is a low possibility of contamination and noise. Instead, these genomes were sequenced from monocultures which have not been subject to environmental stressors or competition. The need to adapt to an extreme environment, like AD, and to outcompete other organisms within the digesters would likely lead to a faster rate of mutation in the *M. soehngenii* MAG. This would ultimately result in the MAG sequences being less similar to reference databases and providing fewer annotations. It is also important to note that the organisms in the NCBI datasets are highly characterised genomes that have multiple, sometimes thousands, of reference genomes that are publicly available and are therefore found within the databases. This suggests that a bias towards the highly characterised genome sequences is likely, as they are also likely to be very similar to the reference sequences in the databases found within the pipeline. Custom databases including microbial data from AD (Becker, *et al.,* 2020) would remove this bias.

Finally, in both metagenomic or genomic sequences, approximately 50% of ORFs (Table 19, Table 26) within these sequences are still unknown. There is only a 12.6% difference between metagenomic and genomic annotations in this paper. This highlights a knowledge gap within biology, the lack of understanding leaves many biological pathways unaccounted for. Efforts to improve this percentage of predictions need to be implemented. An approach to increasing the percentage of predictions of ORFs would be to use a technique such as associative transcriptomics to compare sequences of unknown ORFs together. This would illustrate whether some of the currently unknown ORFs are present, in association with other known proteins. With this information groups of hypothetical ‘pathways’ could be built. Following this, pairwise sequence alignment could be used to indicate similar ORF sequences that could indicate similarities in function and structure.

7.4 The information extrapolated from the outputs of the pipeline is dependent on the user's intended use.

Targeting pipeline outputs to illustrate targeted annotation results is challenging as the pipeline has wide capacity for application. The information discussed within this thesis, regarding the outputs of the pipeline are specific to AD, as this is the overall goal of the research project. The outputs provide breadth rather than depth in the results, to illustrate the capabilities of the pipeline and its wide range of applications.

To generate results with further depth within the context of AD, certain parameters could be implemented prior to the data being inputted into the pipeline. This could be as simple as targeting Functional COGs of interest (regarding metabolism involved in AD process) and taxonomic groups related to microbes (that play a key role in AD, etc.). Another approach would be to develop a bias within custom databases by passing through multiple reference genomes of microorganisms found in AD. This approach has been successfully used by Becker, D. *et al*. A final approach would be to use machine learning to develop a text-mining based search for reference databases that are relevant to the user.

Chapter 8: Limitations

8.1 Limitations in terms of reproducibility.

Reproducibility continues to be an ongoing challenge in pipeline development, this pipeline is subject to some limitations. There are three main limitations.

Pre-existing databases on the Viking Cluster pose an issue for users that do not have access to the Viking Cluster at the University of York, as they will not have access to some of the paths to the databases in the pipeline. This jeopardises the reproducibility of the pipeline, as the databases will have to be installed manually for these users to use the pipeline. However, these databases and tools are easily retrievable from the web and this is commonly accomplished via Unix commands ‘wget’ or ‘curl’.

Additionally, the majority of bioinformatic tools in the pipeline require the prior installation of certain databases before use of the pipeline. This installation will have to be part of the setup, this could be overcome by providing a README file instructing the manual installation required to use the pipeline.

Finally, the file structure is critical to the success of the pipeline. The correct file structure ensures that the databases are accessible for the appropriate bioinformatic tools and that the results are outputted into the correct directory. The use of the workflow management tool, Snakemake, via wildcards in the snakemake script are utilised to lower the limitations with reproducibility from user to user.

Although these points are considered to be limitations, specific installation instructions of the pipeline can be uploaded onto a GitHub (*Github* n.d.) page made for the pipeline (Russell, *et al.,* 2018).

8.2 Limitations regarding pipelines

Although genome annotation pipelines are improving the efficiency of data manipulation and productivity of bioinformatic research etc., there are still challenges and limitations that need to be overcome.

The development of pipelines has significantly improved; however, the process used to annotate genomic data has not drastically changed since these processes were established (Salzberg, 2019). Reference databases for bioinformatic tools are constantly updating, therefore databases rapidly become outdated and are required to be consistently updated (Isserlin, El-Badrawi and Bader, 2011). This results in manual manipulation of the pipeline, removing the automation of the pipeline. In addition, the bioinformatic tools themselves require consistent updates, as new versions of the tools are released, and previous versions may no longer be supported. Out of date tools can cause errors within the pipeline or even restrict use of the bioinformatic tool entirely. This limitation occurred during the development of this pipeline, as EggNOG released an update. Within this update, the developers changed the format of the output file that EggNOG generates. This resulted in multiple parts of the pipeline becoming obsolete, and were required to be re-written for the pipeline to be functional.

Some of the bioinformatic tools utilised in the pipeline only support prokaryotic genome sequences i.e. PROKKA. This is typically a limitation as large knowledge gaps in the datasets will arise. Although this is a limitation, it is not generally a concern for the intended use of the pipeline regarding microbial communities found in AD as a high majority of microbes within AD are prokaryotic (Nguyen, Nguyen and Nghiem, 2019), and only 0.1-1.4% of microbes were found to be eukarya (Matsubayashi, *et al.*, 2017) from AD samples. However efforts to uncover eukaryotes, such as fungi (Schnürer and Schnürer, 2006) and protozoa (Prabhakaran, *et al.,* 2016) with AD are ongoing, by the use of qPCR of RNA and CARD-FISH (Matsubayashi, *et al.*, 2017).

Chapter 9: Conclusion

To conclude, a genome annotation pipeline was developed, using the Snakemake workflow management tool. A selection of functional annotation tools were integrated into the pipeline and the respective results were used to generate a number of tables and figures to visualise the quality of the annotation. A confidence ranking system was successfully devised to compare the results of the bioinformatic tools (EggNOG and KofamScan), where 45.2% of MAG ORFs and an average of 58.7% of ORFs from the NCBI dataset were given a confidence score 1. Identification of high quality protein predictions via these two tools was made possible by the confidence ranking system, as well as, separation of low quality ORFs from the query sequences. Additional annotation of the low qual ORFs via functional and taxonomic analysis ensured more genes are appropriately identified. Finally, the low quality ORFs also undergo an annotation via PROKKA and the proteins of unknown function are annotated as ‘hypothetical’. Alongside this, ORF sequence length is then provided to ensure that the unidentified ORFs are not as a result of a sequencing error.

The capabilities to characterise desired metabolic pathways and the genes involved in these pathways are made possible by the bioinformatic tools in the pipeline. Alongside the figures that are generated, a greater understanding of the role that certain organisms will play in anaerobic digestion was achieved. For example, *Methanothrix soehngenii* is an acetoclastic methanogen. Outputs from the pipeline illustrate the presence and then completeness of the acetoclastic methanogenesis pathway, that allows for the production of methane in AD.

The application and analysis of metagenomic data, allows for the annotation of microorganisms (that would otherwise not be sequenced) by the use of shotgun sequencing (Garza and Dutilh, 2015). The use of genome annotation pipelines, allows for the automated and rapid interpretation of large genomic datasets. This is also applicable to MAGs recovered from AD digesters, to understand the interspecies metabolic pathways that drive the AD process, resulting in the production of methane gas. The methane gas produced can be utilised as a renewable energy source. However, there is significant improvement required to ensure this is an effective process, for production and further use of methane gas from AD. Current applications of AD include wastewater treatment (Wu, *et al.,* 2022), in the conversion of agricultural waste (Jaman, *et al.,* 2022), and more recently in the brewing industry (Herman, *et al.,* 2022). All these processes involve the conversion of organic waste into methane (a high value end product) that significantly reduces contributions to Global GHG emissions.

Limitations with reproducibility and automation are an ongoing challenge that need to be overcome to achieve high levels of productivity from pipelines. Additional limitations regarding the databases required for functional annotation tools continue to be met. The additional limitations are often due to the consistent updates that are required and tools developing a bias of annotation, based on the databases consisting of more well annotated sequences. Finally, databases missing high proportions of reference sequences recognised by the database i.e. 48% of the protein sequences within the KEGG GENES database, do not have an assigned KEGG KO number (Aramaki, *et al.*, 2019).

Chapter 10: Future work

10.1 Improvements to the pipeline

The further development of the pipeline would reduce the limitations regarding reproducibility and automation. For example, access could be ensured to certain databases and bioinformatic tools that were pre-existing on the Viking Cluster (e.g. Seqkit and PROKKA). The addition of output figures and adjustments to pre-existing figures, including the addition of interactive plots, would also further the pipeline's development. Interactive plots would allow for proteins of interest to be easily identified. The effective use of interactive plots is displayed by previous integration into some genome annotation pipelines such as MetaErg (Dong and Strous, 2019). Finally, the use of Blast+ in this pipeline would complete the pipeline.

10.2 Development of a Shiny R application

The development of a Shiny R application would display all outputs generated by the pipeline and improve the accessibility of the outputs for the user. This would be a favourable feature of the pipeline as the skill level of programming and bioinformatics is typically low within the biology research community. Finally, this would also improve the visual aesthetic of the pipeline outputs, encouraging a larger number of users, as it improves usability (David and Glore, 2010).

10.3 Integration of machine learning

Finally, the integration of machine learning to predict expected outputs of types of organisms would improve the productivity of the pipeline. Yip, Cheng, and Gerstein discussed genomic data and its capabilities of automatically recognising patterns via machine, particularly as genomic datasets are often large and unable to undergo manual annotation. Genome annotation pipelines are already integrating machine learning, e.g. DeepAnnotator (Amin, *et al.,* 2018) and LOMETS3 (Zheng, *et al.,* 2022). The application of machine learning would provide examples of the expected output generated by known organisms, by highlighting specific pathways that are involved in that specific MAG's metabolism, or genes of interest within that specific MAG. Currently there are genome annotation pipelines that can manually create a bias for AD microbes as custom databases have been constructed to favour proteins prevalent in microbes in AD (Becker, *et al.,* 2020), as a form of machine learning, thus showing the efficacy of machine learning in AD data.

Abbreviations

MAG	Metagenome-Assembled Genome
ORF	Open Reading Frame
AD	Anaerobic digestion
GWP	Global warming potential
SDG	Sustainable development goals
VFA	Volatile fatty acids
BMP	Biochemical methane potential
NCBI	National centre for biotechnology information
SRB	Sulphate-reducing bacteria
McrA	Methyl coenzyme M reductase A
DGGE	Denaturing gradient gel electrophoresis
KEGG	Kyoto encyclopaedia of genes & genomes
HPC	High performance computing
CWL	Common workflow language
COG	Cluster of orthologous groups
GO	Gene ontology
BLAST+	Basic local alignment tool
LCA	Lowest common ancestor
AEA	Alternative electron acceptor
GUI	Graphical user interface
CSV	Comma-separated variable
rTCA	Reverse Krebs cycle
CARD-FISH	Catalysed reporter deposition-fluorescence in-situ hybridisation
GHG	Greenhouse gas

References

Adobe (2014), 'Adobe Color CC' (Accessed: 4 December 2022).

Alexander, J. *et al.* (2017) ‘Variant Ranker: a web-tool to rank genomic data according to functional significance’, *BMC bioinformatics*, 18(1), p. 341.

Altenhoff, A.M. *et al.* (2016) ‘Standardized benchmarking in the quest for orthologs’, *Nature Methods*, 13(5), pp. 425–430.

Altschul, S.F. *et al.* (1990) ‘Basic local alignment search tool’, *Journal of Molecular Biology*, 215(3), pp. 403–410.

Amin, M.R. *et al.* (2018) ‘DeepAnnotator: genome annotation with deep learning’, in *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, And Health Informatics*. New York, NY, USA: Association for Computing Machinery (BCB ’18), pp. 254–259.

Andrews, S. (2010). FastQC: A Quality Control Tool for High Throughput Sequence Data. [Computer program]. Available http://www.bioinformatics.babraham.ac.uk/projects/fastqc [Accessed 28 November 2022]

Aramaki, T. *et al.* (2019) ‘Kofamkoala: KEGG ortholog assignment based on profile hmm and adaptive score threshold’, *Bioinformatics* , 36(7), Pp. 2251–2252.

Batstone, D.J. *et al.* (2015) ‘Mathematical modelling of anaerobic digestion processes: applications and future needs’, *Reviews in Environmental Science and Biotechnology*, 14(4), pp. 595–613.

Bajpai, P. (2017) ‘Basics of anaerobic digestion process’, In P. Bajpai (Ed.) *Anaerobic Technology in Pulp and Paper Industry*. Singapore: Springer Singapore, pp. 7–12.

Becker, D. *et al.* (2020) ‘A modular metagenomics pipeline allowing for the inclusion of prior knowledge using the example of anaerobic digestion’, *Microorganisms*, 8(5).

Bharti, R. and Grimm, D.G. (2021) ‘Current challenges and best-practice protocols for microbiome analysis’, *Briefings in Bioinformatics*, 22(1), pp. 178–193.

BP (2021) ‘statistical review of world energy’. [Online]. *BP Energy.* Available At: https://www.bp.com/en/global/corporate/energy-economics/statistical-review-of-worldenergy/co2-emissions.html (Accessed: 4 May 2022).

Bueno, E., Pinedo, V. and Cava, F. (2020) ‘Adaptation of Vibrio cholerae to hypoxic environments’, *Frontiers in Microbiology*, 11, p. 739.

Cantalapiedra, C.P. *et al.* (2021) ‘Eggnog-Mapper V2: Functional annotation, orthology assignments, and domain prediction at the metagenomic scale.’ *Molecular Biology and Evolution*.

Camacho, C. *et al.* (2009) ‘BLAST+: Architecture and applications’, *BMC Bioinformatics*, 10, p. 421.

Campanaro, S. *et al.* (2019) ‘The anaerobic digestion microbiome: a collection of 1600 metagenome-assembled genomes shows high species diversity related to methane production’, *bioRxiv*.

Chen, Y., Xia, Z. and Li, H. (2022) ‘Metagenomic comparison of gut communities between hawksbills (Eretmochelys imbricata) and green sea turtles (Chelonia mydas)’, *Archives of Microbiology*, 204(8), p. 450.

Chistiakov, D.A. *et al.* (2014) ‘Intestinal mucosal tolerance and impact of gut microbiota to mucosal tolerance’, *Frontiers In Microbiology*, 5, p. 781.

Ciani, M., Comitini, F. and Mannazzu, I. (2008) ‘Fermentation’, in S.E. Jørgensen and B.D. Fath (eds) *Encyclopedia of Ecology*. Oxford: Academic Press, pp. 1548–1557.

Crameri, F., Shephard, G.E. and Heron, P.J. (2020) ‘The misuse of colour in science communication’, *Nature Communications*, 11(1), p. 5444.

Cruz Ramos, H. *et al.* (2000) ‘Fermentative metabolism of bacillus subtilis: physiology and regulation of gene expression’, *Journal of Bacteriology*, 182(11), pp. 3072–3080.

Darwin and Cord-Ruwisch, R. (2019) ‘Thermodynamics of anaerobic digestion: Mechanism of suppression on biogas production during acidogenesis’, *INMATEH - Agricultural Engineering*, 57(1), pp. 287–301.

David, A. and Glore, P. (2010) The impact of design and aesthetics on usability, credibility, and learning in an online environment. [Online]. *ojdla.com*. Available at: https://ojdla.com/archive/winter134/david_glore134.pdf (Accessed: 5 February 2023).

de Lemos Chernicharo, C.A. (2007) *Anaerobic reactors.* London: IWA Publishing.

Detman, A. *et al.* (2021) ‘Evaluation of acidogenesis products’ effect on biogas production performed with metagenomics and isotopic approaches’, *Biotechnology for Biofuels*, 14(1), p. 125.

Di Tommaso, P. *et al.* (2017) ‘Nextflow enables reproducible computational workflows’, *Nature Biotechnology*, 35(4), pp. 316–319.

Digest of UK Energy Statistics (DUKES) (2021) ‘Digest of UK Energy Statistics (DUKES) 2021: chapters 1-7, annexes A to J and Long-Term Trends Tables’. [Online]. Gov.uk. Available at: https://www.gov.uk/government/statistics/digest-of-uk-energy-statistics-dukes-2021. (Accessed: 4 May 2022).

Dong, X. and Strous, M. (2019) ‘an integrated pipeline for annotation and visualization of metagenomic contigs’, *Frontiers in Genetics*, 10, p. 999.

Finn, R.D. *et al.* (2014) ‘Pfam: The protein families database’, *Nucleic Acids Research*, 42(Database Issue), pp. D222–30.

Förster, A.H. and Gescher, J. (2014) ‘metabolic engineering of escherichia coli for production of mixed-acid fermentation end products’, *Frontiers in Bioengineering and Biotechnology*, 2, p. 16.

Friedrich, M.W. (2005) ‘Methyl-Coenzyme m reductase genes: unique functional markers for methanogenic and anaerobic methane-oxidizing archaea’, *Methods in Enzymology*, 397, pp. 428–442.

Fuchs, S. *et al.* (2007) ‘Anaerobic gene expression in Staphylococcus aureus’, *Journal of Bacteriology*, 189(11), pp. 4275–4289.

Garnier, S., *et al.* (2021). *Viridis - colorblind-friendly color maps for r*. [Computer program]. R package version 0.6.2, Available at: https://sjmgarnier.github.io/viridis/.

Gasc, C. and Peyret, P. (2018) ‘Hybridization capture reveals microbial diversity missed using current profiling methods’, *Microbiome*, 6(1), p. 61.

Gerlach, W. and Stoye, J. (2011) ‘Taxonomic classification of metagenomic shotgun sequences with carma3’, *Nucleic Acids Research*, 39(14), p. E91.

Ghurye, J.S., Cepeda-Espinoza, V. and Pop, M. (2016) ‘Metagenomic assembly: Overview, challenges and applications’, *The Yale Journal of Biology and Medicine*, 89(3), pp. 353–362.

Github (no date). Github. [Online]. Available at: https://github.com/ (Accessed: 6 February 2023).

Graham, E.D., Heidelberg, J.F. and Tully, B.J. (2018) ‘Potential for primary productivity in a globally-distributed bacterial phototroph’, *The ISME journal*, 12(7), pp. 1861–1866.

Guha, T., Fertig, E.J. and Deshpande, A. (2022) ‘Generating colorblind-friendly scatter plots for single-cell data’, *eLife*, 11. Available At: https://doi.org/10.7554/eLife.82128. (Accessed: 11 January 2023).

Hannon, G.J. (2010) *FASTX-Toolkit*. [Computer program]. Available http://hannonlab.cshl.edu/fastx_toolkit. (Accessed: 23 November 2023).

Harirchi, S. *et al.* (2022) ‘Microbiological insights into anaerobic digestion for biogas, hydrogen or volatile fatty acids (vfas): a review’, *Bioengineered*, 13(3), pp. 6521–6557.

Hester J. (2016) A collection of scripts developed to interact with fasta, fastq and sam/bam files. [Computer program]. Available https://github.com/jimhester/fasta_utilities. (Accessed: 23 November 2023).

Herman, T. *et al.* (2022) ‘Comparative fuel yield from anaerobic digestion of emerging waste in food and brewery systems’, *Energies*, 15(4), p. 1538.

Horng, Y.C., Becker, D.F. and Ragsdale, S.W. (2001) ‘Mechanistic studies of methane biogenesis by methyl-coenzyme m reductase: evidence that coenzyme b participates in cleaving the c-s bond of methyl-coenzyme m’, *Biochemistry*, 40(43), pp. 12875–12885.

Huerta-Cepas, J. *et al.* (2019) ‘eggNOG 5.0: A hierarchical, functionally and phylogenetically annotated orthology resource based on 5090 organisms and 2502 viruses’, *Nucleic Acids Research*, 47(D1), pp. D309–D314.

Huerta-Cepas, J. *et al.* (2017) ‘Fast genome-wide functional annotation through orthology assignment by eggnog-mapper’, *Molecular biology and evolution*, 34(8), pp. 2115–2122.

Hyatt, D. *et al.* (2010) ‘Prodigal: Prokaryotic gene recognition and translation initiation site identification’, *BMC Bioinformatics*, 11, p. 119.

Jetten, M.S.M., Stams, A.J.M. and Zehnder, A.J.B. (1992) ‘Methanogenesis from acetate: A comparison of the acetate metabolism in methanothrix soehngenii and methanosarcina spp’, *FEMS Microbiology Reviews*, 8(3-4), pp. 181–197.

Isserlin, R., El-Badrawi, R.A. and Bader, G.D. (2011) ‘The biomolecular interaction network database in psi-mi 2.5’, *The Journal of Biological Databases and Curation*, 2011, p. Baq037.

Jalili, V. *et al.* (2020) ‘The galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2020 update’, *Nucleic Acids Research*, 48(W1), pp. W395–W402.

Jaman, K. *et al.* (2022) ‘Anaerobic digestion, codigestion of food waste, and chicken dung: correlation of kinetic parameters with digester performance and on-farm electrical energy generation potential’, *Fermentation*, 8(1), p. 28.

Jiang, C. *et al.* (2018) ‘Monitoring foaming potential in anaerobic digesters’, *Waste Management* , 75, pp. 280–288.

Jones, P. *et al.* (2014) ‘InterProScan 5: Genome-scale protein function classification’, *Bioinformatics* , 30(9), pp. 1236–1240.

Kanehisa, M. *et al.* (2016) ‘KEGG as a reference resource for gene and protein annotation’, *Nucleic Acids Research*, 44(D1), pp. D457–62.

Kampers, L.F.C. *et al.* (2021) ‘A metabolic and physiological design study of pseudomonas putida kt2440 capable of anaerobic respiration’, *BMC microbiology*, 21(1), p. 9.

Kanehisa, M. and Sato, Y. (2020) ‘KEGG mapper for inferring cellular functions from protein sequences’, *Protein Science: A Publication of the Protein Society*, 29(1), pp. 28–35.

Kanehisa, M., Sato, Y. and Morishima, K. (2016) ‘Blastkoala and ghostkoala: kegg tools for functional characterization of genome and metagenome sequences’, *Journal of Molecular Biology*, 428(4), pp. 726–731.

Kirkegaard, R.H. *et al.* (2017) ‘The impact of immigration on microbial community composition in full-scale anaerobic digesters’, *Scientific Reports*, 7(1), p. 9343.

Köster, J. and Rahmann, S. (2012) ‘Snakemake--a scalable bioinformatics workflow engine’, *Bioinformatics* , 28(19), pp. 2520–2522.

Lagesen, K. *et al.* (2007) ‘RNAmmer: Consistent and rapid annotation of ribosomal rna genes’, *Nucleic Acids Research*, 35(9), pp. 3100–3108.

Laslett, D. and Canback, B. (2004) ‘ARAGORN, A program to detect trna genes and tmrna genes in nucleotide sequences’, *Nucleic Acids Research*, 32(1), pp. 11–16.

Li, H. (2012) ‘seqtk toolkit for processing sequences in fasta/q formats’, *GitHub* [Computer program]. Available https://github.com/lh3/seqtk. [Accessed 23 November 2023].

Lim, J.W. *et al.* (2020) ‘The microbiome driving anaerobic digestion and microbial analysis’, *Advances in Bioenergy*, 5, p. 1.

Garza, D.R. and Dutilh, B.E. (2015) ‘From cultured to uncultured genome sequences: metagenomics and modeling microbial ecosystems’, *Cellular and Molecular Life Sciences: CMLS*, 72(22), pp. 4287–4308.

Liu, Y. (2010) ‘Taxonomy of methanogens’, in K.N. Timmis (ed.) *Handbook of Hydrocarbon and Lipid Microbiology*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 547–558.

Lyu, Z. and Liu, Y. (2019) ‘Diversity and taxonomy of methanogens’, in A.J.M. Stams and D.Z. Sousa (eds) *Biogenesis of Hydrocarbons*. Cham: Springer International Publishing, pp. 19–77.

Ma, J. *et al.* (2013) ‘A simple methodology for rate-limiting step determination for anaerobic digestion of complex substrates and effect of microbial community ratio’, *Bioresource Technology*, 134, pp. 391–395.

Mahmood, Q. *et al.* (2006) ‘the rate-limiting step in anaerobic digestion in the presence of phosphine’, *Toxicology and Industrial Health*, 22(4), pp. 165–172.

Matsubayashi, M. *et al.* (2017) ‘Phylogenetic diversity and in situ detection of eukaryotes in anaerobic sludge digesters’, *PloS one*, 12(3), p. e0172888.

Meegoda, J.N. *et al.* (2018) ‘A review of the processes, parameters, and optimization of anaerobic digestion’, *International Journal of Environmental Research and Public Health*, 15(10). Available at: https://doi.org/10.3390/ijerph15102224.

Morris, R. *et al.* (2014) ‘Methyl coenzyme m reductase (mcra) gene abundance correlates with activity measurements of methanogenic h_2 /co_2 -Enriched Anaerobic Biomass’, *Microbial Biotechnology*, 7(1), pp. 77–84.

Mukhuba, M. *et al.* (2018) ‘Comparative assessment of bio-fertiliser quality of cow dung and anaerobic digestion effluent’, *Cogent Food & Agriculture*, 4(1), p. 1435019.

National Library of Medicine (US), National Center for Biotechnology Information (1988). *Nucleotide* [Online]. NCBI. Available at: https://www.ncbi.nlm.nih.gov/nuccore/NZ_CP103783.1 (Accessed 08 February 2023).

Nawrocki, E.P. and Eddy, S.R. (2013) ‘Infernal 1.1: 100-fold faster rna homology searches’, *Bioinformatics* , 29(22), pp. 2933–2935.

Nayfach, S. *et al.* (2021) ‘CheckV assesses the quality and completeness of metagenome assembled viral genomes’, *Nature Biotechnology*, 39(5), pp. 578–585.

Neuwirth, E. and Neuwirth, M.E. (2014) ‘Package “RColorBrewer”’. *ColorBrewer Palettes* [Computer Program]. Available at: https://mirror.niser.ac.in/cran/web/packages/RColorBrewer/RColorBrewer.pdf. (Accessed 18 January 2023).

Nguyen, L.N., Nguyen, A.Q. and Nghiem, L.D. (2019) ‘Microbial community in anaerobic digestion system: progression in microbial ecology’, in X.-T. Bui et al. (eds) *Water and Wastewater Treatment Technologies*. Singapore: Springer Singapore, pp. 331–355.

Nguyen, T.T. *et al.* (2021) ‘Gut microbiome in schizophrenia: altered functional pathways related to immune modulation and atherosclerotic risk’, *Brain, Behaviour, and Immunity*, 91, pp. 245–256.

Nissen, J.N. *et al.* (2021) ‘Improved metagenome binning and assembly using deep variational autoencoders’, *Nature Biotechnology*, 39(5), pp. 555–560.

Olson, N.D. *et al.* (2019) ‘Metagenomic assembly through the lens of validation: recent advances in assessing and improving the quality of genomes assembled from metagenomes’, *Briefings in Bioinformatics*, 20(4), pp. 1140–1150.

Ou, J. (2021) *colorblindness Guide* [Computer program]. Available at: https://cran.rproject.org/web/packages/colorBlindness/vignettes/colorBlindness.html#References. (Accessed 18 January 2023).

Pan, X. *et al.* (2021) ‘Deep insights into the network of acetate metabolism in anaerobic digestion: focusing on syntrophic acetate oxidation and homoacetogenesis’, *Water Research*, 190, p. 116774.

Parks, D.H. *et al.* (2015) ‘CheckM: Assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes’, *Genome Research*, 25(7), pp. 1043–1055.

Patel, G.B. and Sprott, G.D. (1990) ‘Methanosaeta concilii gen. nov. sp. nov. (“methanothrix concilii”) and methanosaeta thermoacetophila nom. rev., comb. nov.?’, *International Journal of Systematic Bacteriology*, 40, pp. 79–82.

Perkel, J.M. (2019) ‘Workflow systems turn raw data into scientific knowledge’, *Nature*, 573(7772), pp. 149–150.

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. *Microbial Genomics’*.

Prabhakaran, P. *et al.* (2016) ‘Community dynamics and significance of anaerobic protozoa during biomethanation of lignocellulosic waste’, *Renewable Energy*, 98, pp. 148–152.

Ruiz-Perez, C.A., Conrad, R.E. and Konstantinidis, K.T. (2021) ‘MicrobeAnnotator: A userfriendly, comprehensive functional annotation pipeline for microbial genomes’, *BMC Bioinformatics*, 22(1), p. 11.

Russell, P.H. *et al.* (2018) ‘A large-scale analysis of bioinformatics code on github’, *PloS One*, 13(10), p. e0205898.

S&P Global Market Intelligence (no date). Available at: https://www.spglobal.com/marketintelligence/en/ (Accessed: 6 February 2023).

Salzberg, S.L. (2019) 'Next-generation genome annotation: we still struggle to get it right', *Genome Biology*, 20(1), p. 92.

Schäffler, H. and Breitrück, A. (2018) 'Clostridium difficile - from colonization to infection', *Frontiers in Microbiology*, 9, p. 646.

Schoch, C.L. *et al.* (2020) 'NCBI taxonomy: A comprehensive update on curation, resources and tools'. [Database]. *The Journal of Biological Databases and Curation*. Available at: https://doi.org/10.1093/database/baaa062. (Accessed 11 March 2022)

Schilling, O. *et al.* (2007) 'Transcriptional and metabolic responses of bacillus subtilis to the availability of organic acids: transcription regulation is important but not sufficient to account for metabolic adaptation', *Applied and Environmental Microbiology*, 73(2), pp. 499–507.

Schnürer, A. and Schnürer, J. (2006) 'Fungal survival during anaerobic digestion of organic household waste', *Waste Management* , 26(11), pp. 1205–1211.

Seemann, T. (2014) 'Prokka: Rapid prokaryotic genome annotation', *Bioinformatics* , 30(14), pp. 2068–2069.

Servin, B., de Givry, S. and Faraut, T. (2010) 'Statistical confidence measures for genome maps: application to the validation of genome assemblies', *Bioinformatics* , 26(24), pp. 3035–3042.

Shell Energy (2021) 'Sustainability Report 2021'. [Online]. *Shell Energy.* Available at: https://reports.shell.com/sustainability-report/2021/. (Accessed 20 March 2022).

Shen, W. *et al.* (2016) 'SeqKit: A cross-platform and ultrafast toolkit for fasta/q file manipulation', *PloS One*, 11(10), p. E0163962.

Shirley, M.D. *et al.* (2015) Efficient 'pythonic' access to fasta files using pyfaidx. *PeerJ PrePrints.*

Smith, C. *et al.* (2021) 'The earth's energy budget, climate feedbacks, and climate sensitivity supplementary material', *Climate Change and Environmental Sustainability* [Preprint].

Sun, Y., Kokko, M. and Vassilev, I. (2023) 'Anode-assisted electro-fermentation with bacillus subtilis under oxygen-limited conditions', *Biotechnology for Biofuels and Bioproducts*, 16(1), p. 6.

Strous, M. *et al.* (2012) 'The binning of metagenomic contigs for microbial physiology of mixed cultures', *Frontiers in Microbiology*, 3, p. 410.

Suzuki, S. *et al.* (2014) 'GHOSTX: an improved sequence homology search algorithm using a query suffix array and a database suffix array', *PloS One*, 9(8), p. E103833.

Tatusova, T. *et al.* (2016) ‘NCBI prokaryotic genome annotation pipeline’, *Nucleic Acids Research*, 44(14), pp. 6614–6624.

Teufel, F. *et al.* (2022) ‘SignalP 6.0 predicts all five types of signal peptides using protein language models’, *Nature Biotechnology*, 40(7), pp. 1023–1025.

Trachana, K. *et al.* (2014) ‘A phylogeny-based benchmarking test for orthology inference reveals the limitations of function-based validation’, *PloS One*, 9(11), p. E111122.

UN General Assembly (2015) *‘Transforming our world : The 2030 Agenda for Sustainable Developments’*, A/RES/70/1’. [Online]. Available at: https://www.refworld.org/docid/57b6e3e44.html (Accessed 6 February 2023).

United Nations / Framework Convention on Climate Change (2015) Adoption of the Paris Agreement, 21st Conference of the Parties, Paris: United Nations.

Urbanowicz, R.J. *et al.* (2018) ‘Benchmarking relief-based feature selection methods for bioinformatics data mining’, *Journal of Biomedical Informatics*, 85, pp. 168–188.

Van Alst, N.E. *et al.* (2007) ‘Nitrate sensing and metabolism modulate motility, biofilm formation, and virulence in pseudomonas aeruginosa’, *Infection and Immunity*, 75(8), pp. 3780–3790.

Vlasova, A. *et al.* (2021) ‘FA-nf: A functional annotation pipeline for proteins from nonmodel organisms implemented in nextflow’, *Genes*, 12(10). Available at: https://doi.org/10.3390/genes12101645.

Vrancken, G. *et al.* (2019) ‘Synthetic ecology of the human gut microbiota’, *Nature Reviews. Microbiology*, 17(12), pp. 754–763.

Wang, Z. *et al.* (2021) ‘Accelerating anaerobic digestion for methane production: potential role of direct interspecies electron transfer’, *Renewable and Sustainable Energy Reviews*, 145, p. 111069.

Westlund, Å.D., Hagland, E. and Rothman, M. (1998) ‘Foaming in anaerobic digesters caused by microthrix parvicella’, *Water Science and Technology: A Journal of the International Association on Water Pollution Research*, 37(4), pp. 51–55.

Wickramarachchi, A. and Lin, Y. (2022) ‘Binning long reads in metagenomics datasets using composition and coverage information’, *Algorithms for Molecular Biology: AMB*, 17(1), p. 14.

Wood, D.E., Lu, J. and Langmead, B. (2019) ‘Improved metagenomic analysis with kraken 2’, *genome biology*, 20(1), p. 257.

Wood, D.E. and Salzberg, S.L. (2014) ‘Kraken: Ultrafast metagenomic sequence classification using exact alignments’, *Genome Biology*, 15(3), p. R46.

Wratten, L., Wilm, A. and Göke, J. (2021) ‘Reproducible, scalable, and shareable analysis pipelines with bioinformatics workflow managers’, *Nature Methods*, 18(10), pp. 1161–1168.

Wu, Q. *et al.* (2022) ‘Effects of antibiotics on anaerobic digestion of sewage sludge: performance of anaerobic digestion and structure of the microbial community’, *The Science of the Total Environment*, 845, p. 157384.

Yang, Y., Gilbert, D. and Kim, S. (2010) ‘Annotation confidence score for genome annotation: a genome comparison approach’, *Bioinformatics* , 26(1), pp. 22–29.

Yip, K.Y., Cheng, C. and Gerstein, M. (2013) ‘Machine learning and genome annotation: a match meant to be?’, *Genome Biology*, 14(5), p. 205.

You, S., Sonne, C. and Ok, Y.S. (2020) ‘COVID-19’s unsustainable waste management’, *Science*, 368(6498), p. 1438.

Yu, H.Q. and Fang, H.H.P. (2003) ‘Acidogenesis of gelatin-rich wastewater in an upflow anaerobic reactor: influence of ph and temperature’, *Water Research*, 37(1), pp. 55–66.

Zhang, J. *et al.* (2017) ‘Metagenomic insight into the microbial networks and metabolic mechanism in anaerobic digesters for food waste by incorporating activated carbon’, *Scientific Reports*, 7(1), p. 11293.

Zhang, L. *et al.* (2019) ‘Bioinformatics analysis of metagenomics data of biogas-producing microbial communities in anaerobic digesters: A review’, *Renewable and Sustainable Energy Reviews*, 100, pp. 110–126.

Zhang, L. *et al.* (2021) ‘Advances in metagenomics and its application in environmental microorganisms’, *Frontiers in microbiology*, 12, p. 766364.

Zhang, Q. *et al.* (2021) ‘Gut microbiome composition associated with major depressive disorder and sleep quality’, *Frontiers in Psychiatry / Frontiers Research Foundation*, 12, p. 645045.

Zheng, W. *et al.* (2022) ‘LOMETS3: Integrating deep learning and profile alignment for advanced protein template recognition and function annotation’, *Nucleic acids research*, 50(W1), pp. W454–64.

Supplementary

Scripts written to produce the pipeline from snakemake, Python can be found in a Github repository https://github.com/louisegrimble/MRes-pipeline. R scripts for the data visualisation can also be found within this repository.

Supplementary data (S.1-5) referred to in the main text can be found in a Github repository https://github.com/louisegrimble/MRes-pipeline. All additional files generated by the pipeline can be found in the Github repository.