Utilising Nanopore technology for interactive real-time metagenomics

Ned Peel

A thesis submitted to the University of East Anglia
for the degree of Doctor of Philosophy (PhD)

Earlham Institute

October 2021

Acknowledgements

Firstly, I would like to thank my primary supervisor, Dr Richard Leggett, for giving me the opportunity to work with this exciting new technology, and for his excellent guidance, encouragement, and support throughout. I am also grateful to my secondary supervisor, Dr Douglas Yu, and team members, Dr Matthew Clark and Dr Robert Davey, for the time they gave to my work and for their advice, ideas, and constructive feedback.

I am grateful to the members of the Leggett group, past and present, for their support and friendship. I would like to give special thanks to Dr Darren Heavens for sharing his knowledge and providing laboratory guidance and assistance.

I would also like to thank the Earlham Institute for the excellent facilities they have provided that have enabled me to carry out this research. I am also grateful to the Research Computing department of the Norwich Bioscience Institutes for providing the computing infrastructure required for this work.

I am grateful to the UKRI Biotechnology and Biological Sciences Research Council Norwich Research Park Biosciences Doctoral Training Partnership for supporting this work and providing my studentship.

Finally, I would like to thank my family and my partner, Elizabeth Bankes-Jones, for their love and support.

Publications

Peel, N., Dicks, L. V., Clark, M. D., Heavens, D., Percival-Alwyn, L., Cooper, C., Davies, R. G., Leggett, R. M. and Yu, D. W. (2019). Semi-quantitative characterisation of mixed pollen samples using MinION sequencing and Reverse Metagenomics (RevMet). *Methods in Ecology and Evolution*, 10(10), 1690-1701.

Leggett, R. M., Alcon-Giner, C., Heavens, D., Caim, S., Brook, T. C., Kujawska, M., Martin, S., Peel, N., Acford-Palmer, H., Hoyles, L., Clarke, P., Hall, L. J. and Clark, M. D. (2020). Rapid MinION profiling of preterm microbiota and antimicrobial-resistant pathogens. *Nature Microbiology*, 5(3), 430-442.

Reddington, K., Eccles, D., O'Grady, J., Drown, D. M., Hansen, L. H., Nielsen, T. K., Ducluzeau, A.-L., Leggett, R. M., Heavens, D., Peel, N., Snutch, T. P., Bayega, A., Oikonomopoulos, S., Ragoussis, J., Barry, T., van der Helm, E., Jolic, D., Richardson, H., Jansen, H., Tyson, J. R., Jain, M. and Brown, B. L. (2020). Metagenomic analysis of planktonic riverine microbial consortia using nanopore sequencing reveals insight into river microbe taxonomy and function. *Gigascience*, 9(6).

Peel, N., Martin, S., Heavens, D., Davey, R. P., Yu, D. W., Clark, M. D. and Leggett, R. M. (2022). MARTi: a real-time analysis and visualisation tool for nanopore metagenomics. *In preparation*.

Abstract

Nanopore sequencing technology has the potential to revolutionise metagenomics by providing long reads, which can improve taxonomic classification and assembly contiguity, near real-time analysis, enabling rapid results and improved sequencing efficiency, and portability, allowing sequencing in the field. However, the full potential of these features is largely unrealised due to the lack of available tools and methods. In this thesis, we report on tools and analysis methods that facilitate the use of nanopore sequencing technology for metagenomics and real-time analysis.

Applying metagenomics to samples containing a mix of eukaryote species, such as bee-collected pollen, is challenging due to lack of available reference genomes. This thesis presents a new method, RevMet (Reverse Metagenomics), for semi-quantitative characterisation of mixed eukaryote samples without the need for complete reference genomes. Instead, each reference species is represented by a low-cost genome skim. The short-read reference skims are mapped to the long nanopore query reads to individually classify them, which is the reverse of the standard metagenomic approach of mapping reads to (assembled) references.

Recognising the need for an open-source software tool for real-time analysis and visualisation of metagenomic sequencing data, we developed MARTi (Metagenomic Analysis in Real-Time). MARTi provides a rapid, lightweight web interface that allows users to view community composition and identify antimicrobial resistance genes in real time. MARTi consists of two main parts, the Engine and the GUI, and can be configured in multiple ways to suit the needs of the user. We demonstrate MARTi on live nanopore sequencing runs - firstly, using a mock gut community and, secondly, using clinical faecal gut microbiome samples taken from patients suffering from liver disease.

Contents

List of Tables

List of Figures

1 Chapter 1 – Introduction

1.1 Metagenomics

Microorganisms have colonised almost every natural environment on Earth, they regulate global biogeochemical cycling, have a huge impact on the biosphere, influence host health, and are the most abundant and diverse organisms of all three domains of life. Until recently, only a small fraction of this diversity could be characterised as traditional microbiology relies on culture-based methods and the vast majority (>99%) of microorganisms are unculturable with current techniques (Amann et al. 1995). The emergence of high-throughput sequencing has led to new ways of studying the microbial world, revealing previously hidden diversity and functional ability of complex microbiomes. These methods include metabarcoding, which involves the amplification and sequencing of marker genes which must contain conserved regions (for primer design) flanking variable regions which when sequenced are taxonomically informative e.g. ribosomal RNA genes, and metagenomics, the untargeted sequencing of all genomes present in a sample (Driscoll et al. 2017).

In contrast to targeted metabarcoding approaches, metagenomics uses genome-wide shotgun sequencing to capture a wealth of genomic information from across the genomes of all members in complex communities. As a result, metagenomics not only offers taxonomic profiling with greater discriminatory power, but also functional insights, and potentially whole genomes. Metagenomics has a wide range of applications that includes facilitating the discovery of new enzymes and biomolecules with potential applications in industry and medicine (Wilson and Piel 2013; Ferrer et al. 2016), providing a better understanding of the composition and functional ecology of different environmental communities (Grossart et al. 2020), and identification of pathogens and their antimicrobial resistance genes in clinical samples (Chiu and Miller 2019; De 2019).

1.2 Third-generation sequencing

The rapid development and expansion of the metagenomic field was largely driven by advances in sequencing technologies and associated computational analysis (Escobar-Zepeda et al. 2015). Next-generation sequencing (NGS) technologies, such as Illumina’s sequencing by synthesis (SBS), greatly reduced sequencing costs due to massive increases in throughput. Consequently, metagenomic sequencing projects could be carried out at a greater scale and depth than possible with Sanger sequencers (Karsenti et al. 2011). Nevertheless, due to the short read lengths (35 - 300 bp) produced by Illumina’s sequencing platforms, the information contained in individual sequences is limited. As a result, reads can be difficult to correctly classify and for many applications must first be assembled into longer contiguous sequences. Assembly of such short sequences is computationally challenging and as they are unable to span intra- and intergenomic repeats, can result in highly fragmented metagenomic assemblies (Somerville et al. 2019; Ayling et al. 2020).

Third-generation sequencing technologies, such as the platforms developed by Pacific Biosciences (PacBio) and Oxford Nanopore Technologies (ONT), can produce extremely long reads, i.e., multi-kilobase and even up to multi-megabase for ONT. Although they have lower per base accuracy than short reads, longer reads contain more information per read, and can overcome many of the known problems associated with short-read metagenomics, especially when the sequencing depth is high enough to allow error correction (Koren et al. 2012; Somerville et al. 2019). Long reads have increased taxonomic classification accuracy and can simplify the challenge of metagenomic assembly (Pearman et al. 2020; Cusco et al. 2021). Furthermore, these platforms can sequence native molecules, thus eliminating amplification bias and enabling detection of base modifications (Amarasinghe et al. 2020). Despite these advantages, long-read metagenomics is still in its infancy and there are fewer tools available for assembly and classification of metagenomic sequences from third generation platforms (Driscoll et al. 2017; Kolmogorov et al. 2020).

1.3 Nanopore sequencing

1.3.1 The MinION

ONT's MinION was the world's first commercially available nanopore-based DNA sequencer. Due to its small size and ability to generate ultra-long reads, the USB-powered device is particularly attractive for metagenomic analysis and in-field sequencing. Nanopore sequencing is being applied in metagenomics in a number of different ways, including: assembly of complete bacterial genomes from complex microbiomes (Moss et al. 2020), pathogen identification and clinical diagnosis (Charalampous et al. 2019), and using the DNA methylation data for binning metagenomic contigs and linking mobile genetic elements to their host genomes (Tourancheau et al. 2021). Furthermore, the MinION has been deployed in a number of different environments for in-field sequencing, including the Antarctic dry valleys (Johnson et al. 2017), cloud forests in Tanzania (Menegon et al. 2017), and even on the International Space Station (Castro-Wallace et al. 2017).

1.3.2 ONT's sequencing platforms

Since first becoming publicly available in April 2014 when the MinION Access Programme (MAP) was launched, ONT have continually improved their sequencing chemistry and basecalling software resulting in substantial increases in throughput and accuracy. Early adopters of the MinION, using the first available chemistry (R6), reported typical flow cell yields of 100s of megabases with median accuracy of 61.6% to 71.5% based on mapping back to reference genomes (Ashton et al. 2015; Laver et al. 2015). However, with more recent chemistries and software the MinION regularly produces over 10 gigabases per flow cell with median accuracy of greater than 90% (Urban et al. 2021; Wu et al. 2021). ONT has also expanded its range of sequencing hardware in order to cater for projects of various sizes and currently consists of the following:

- MinION (Mk1B and Mk1C) – There are currently two versions of the MinION available, the base model, Mk1B, and a newer model that has integrated basecalling hardware and a screen, Mk1C. Both devices take a single MinION flow cell, which can produce up to 30 gigabases of sequencing data.
- GridION (Mk1) – A benchtop device with integrated real-time basecalling hardware and capacity to run five MinION flow cells either individually or at the same time, producing up to 5 x 30 Gb, 150 Gb total yield.
- PromethION (P24 and P48) – Designed for large-scale sequencing, PromethION comes in two models, P24 and P48, that have capacity for 24 and 48 PromethION flow cells respectively. Each flow cell is capable of producing up to 200 Gb, giving total potential yields of 4.8 Tb and 9.6 Tb.
- Flongle – A flow cell dongle, or adapter, that allows cheap, low throughput, Flongle flow cells to be run on a MinION or GridION. Produces up to 1.8 Gb per flow cell.

1.3.3 Strand sequencing technology

Unlike Illumina's SBS technology, nanopore sequencing does not require DNA template amplification or the use of modified fluorescent bases. Instead, ONT's technology, known as 'Strand sequencing', allows direct detection of native polynucleotides, DNA or RNA, passing through protein nanopores embedded in an electrically resistant membrane (Figure 1.1). A voltage is applied across the membrane, causing ions to flow through the embedded pores. During library preparation, modified helicase enzymes, known as ‘Motor proteins’, are added to the ends of each DNA/RNA molecule. These proteins unzip the double helices and feed single stranded molecules through the pores at a steady rate. As strands pass through the pores, sensors detect the disruptions in ionic current across each pore caused by the traversing nucleotide sequences. The current measurements, referred to as ‘squiggles’, are computationally interpreted to produce basecalled sequences.

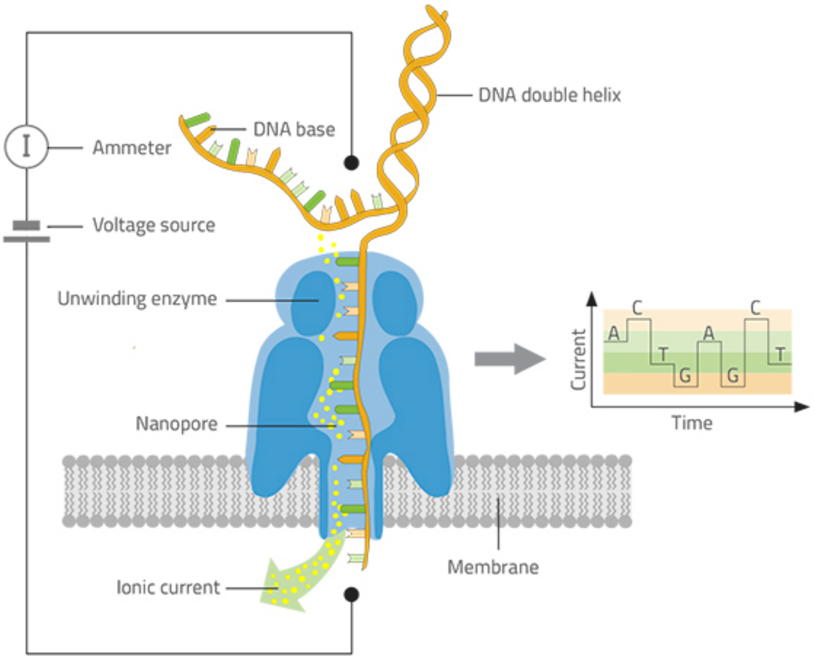


Figure 1.1 Illustration of ONT’s strand sequencing technology. A voltage applied across the membrane causes ions to flow through the embedded nanopores. Single stranded DNA is fed through the pore at a controlled rate by the motor protein. The combination of nucleotides present in the pore causes characteristic disruptions in the electrical current. The ionic current measurements, known as squiggles, are converted into basecalled sequences. Figure from Kerstin Göpfrich, *Science in School* (https://www.scienceinschool.org/article/2018/decoding-dna-pocket-sized-sequencer).

1.3.4 Library preparation

ONT have developed an array of different library preparation kits and protocols for a wide range of DNA and RNA sequencing applications. The most applicable kits for shotgun metagenomics are the Ligation Sequencing Kit (LSK) and the Rapid Sequencing Kit (RSK). These kits are amplification free and therefore allow direct sequencing of native DNA, eliminating the potential for PCR bias and enabling the detection of base modifications. The LSK is optimised for throughput and involves fragmentation of DNA to desired length (optional, but can improve yield), followed by repair and dA-tailing of ends, and finally ligation of sequencing adapters. On the other hand, the RSK is optimised for simplicity and speed (i.e., can be completed in as little as 10 minutes), relying on transposase to randomly fragment DNA and simultaneously add sequencing adapters. There is also a lyophilized version of the RSK, called the Field Sequencing Kit, that can be stored at temperatures of up to 30 °C for one month making it ideal for use in the field where access to cold storage may be limited.

1.3.5 Basecalling

The raw ionic current measurements are converted to DNA or RNA bases, and several types of modified bases, by ONT's Guppy software. The tool can be run from the command line for post-sequencing basecalling, but is also integrated into ONT's sequencing instrument software, MinKNOW, for real-time basecalling and demultiplexing. Real-time basecalling is made up of several processes that are executed one by one: First, the ionic current measurements from the sequencing device are collected by MinKNOW and processed into a read; the raw read signal is transformed into basecalls using models based on a Recurrent Neural Network (RNN); and finally, the basecalled reads are written into FASTQ files, with a default of 4000 reads per file.

Guppy can be run on both Central Processing Units (CPUs) and Graphics Processing Units (GPUs). However, it is optimised for certain NVIDIA GPUs and can perform several orders of magnitude faster running on one of these GPUs vs a standard desktop CPU.

Guppy offers three different basecalling models: Fast model, designed to keep up with real-time data generation on ONT’s sequencing platforms; High accuracy (HAC) model, provides higher accuracy than the Fast model, but is 5 to 8 times more computationally intensive; and Super accurate (sup) model, that provides even higher read accuracy than the HAC model, but is roughly 3 times more intensive.

1.3.6 Real-time analysis

A key advantage of ONT’s sequencing platforms is their potential for real-time analysis. The devices stream the electrical current data from individual nanopores as DNA molecules pass through them, enabling basecalling and other analyses to occur whilst sequencing is still in progress. This capability can be exploited for time-critical applications, such as rapid infection diagnosis and real-time environmental monitoring. Proof-of-concept studies have demonstrated real-time pathogen identification in clinical samples including prosthetic joints and preterm baby gut microbiomes (Sanderson et al. 2018; Leggett et al. 2020).

Real-time analysis also allows sequencing decisions to be made on the fly. For example, with the technique ONT calls ‘Run until’, sequencing can be ended early if enough data has been collected to meet the goal of the run, such as a certain sequencing depth reached for a specific region of interest, or species of interest (Payne et al. 2021).

Another real-time utility of ONT’s platforms is ‘Adaptive sequencing’, which allows users to enrich or deplete target DNA during a run. DNA molecules are ejected by reversing the voltage across individual pores, freeing them up for new molecules. This technique has been demonstrated with varying success in both squiggle space, by pattern matching live ‘squiggles’ to synthesized reference squiggles of the target sequence, and base space, which relies on real-time basecalling and alignment to reference sequences (Loose et al. 2016; Martin et al. 2021; Payne et al. 2021).

1.3.7 Limitations

With the MinION Mk1B starter pack costing just £800 (includes the sequencer, a flow cell, a flow cell wash kit, and a library preparation kit), nanopore sequencing is accessible to almost any research group in the world, enabling them to generate their own long-read data and access it in real-time. Nonetheless, nanopore sequencing currently has several limitations.

Perhaps of most concern in many applications is the error rate. The current nanopore error rate (typically < 5%) is much higher than that of Illumina (~0.1%) and is enough to obscure differences between highly similar sequences. Nanopore error rate can be expected to reduce over time due to upgrades in hardware, chemistry, and software. Some of the ways ONT hopes to improve accuracy in the near future include engineering new pores with improved signal-to-noise ratio, developing a method of re-reading native molecules, and improving the basecalling algorithm.

Although read length is one of nanopore sequencing's biggest strengths, further increases in average read length would facilitate improved assembly and taxonomic classification accuracy. Recall (the ratio of correctly classified reads to all reads) for long nanopore reads (4000 bp) has been shown to be equal to or higher than even the longest Illumina reads (300 bp) regardless of kingdom or classification method. For animals and plants, recall improves almost three-fold as read length increases from 300 bp to 4000 bp (Pearman et al. 2020). Generating long metagenomic reads from all organisms within a community will require improved DNA extraction methods and library preparation protocols. Imposing a minimum read length filter before classification is also recommended.

Another potential limitation of nanopore sequencing is the relatively high input requirements for producing a library. As PCR amplification is unsuitable for very long DNA fragments and does not preserve base modifications, PCR-free library preparation is required for many applications. As a result, the required DNA input can be in the multi-microgram range for long-read libraries.

Finally, as a newer sequencing technology, there are fewer user-friendly bioinformatics tools available to nanopore users. As a result, the bioinformatic and computational skill required to analyse the data is relatively high. New easy-to-use tools and methods would further increase the accessibility of nanopore sequencing.

1.4 Analysis tools

Together, the long reads and real-time analysis potential provided by nanopore sequencing could revolutionise metagenomics by improving classification accuracy, metagenomic assembly, sequencing efficiency, and by reducing the time to result. However, the full potential of these features is largely unrealised as the analysis tools and pipelines lag behind the hardware developments. As a result, metagenomic analysis of nanopore data often relies on methods and tools that have been optimised for high-quality short reads.

1.4.1 Classification

One of the main challenges in the field of metagenomics has been the development of computational methods to accurately classify reads to taxa. Strategies for read classification can be divided into two main categories: alignment-based, such as BLAST and minimap2; and alignment-free, which includes Kraken, Kraken2, Centrifuge, CLARK, MetaMaps and Bracken.

BLAST (basic local alignment and search tool) is one of the most popular tools for sequence alignment and is considered by some to be the gold standard (Altschul et al. 1990). Although not specifically designed for metagenomics, it is easily applied to this problem and remains one of the best methods available. BLAST, and other alignment-based tools, can be very accurate and can output useful information such as genomic locations and alignment qualities for individual reads. Alignment-based tools are computationally intensive and are typically slower than alignment-free methods. Efforts have been made to increase the speed of alignment tools by developing both more efficient algorithms and hardware-accelerated implementations of existing tools. For example, DIAMOND, an alternative algorithm to BLASTX, is capable of aligning translated nucleotide queries to protein databases at twenty-thousand times the rate of the original tool (Buchfink et al. 2015).

The ever-increasing throughput of sequencing technologies coupled with the exponential growth of the number of available microbial genomes has led to massive increases in the number of comparisons that need to be performed during classification. Alignment-free tools such as Kraken and Centrifuge have been developed for faster and more efficient metagenomic classification and abundance estimation (Wood and Salzberg 2014; Kim et al. 2016). To classify a sequence, Kraken searches for exact matches in a database for each k-mer in the query sequence. Each k-mer is mapped to the lowest common ancestor (LCA) of the genomes that contain that k-mer. The lowest level taxon with k-mer assignments in the root-to-leaf (RTL) path with the highest sum of mapped k-mers is the classification used for the query sequence. With short, accurate, reads, Kraken can be both accurate and fast, achieving low-level resolution and classifying millions of reads per minute (Wood and Salzberg 2014). However, Kraken’s exact k-mer matching algorithm requires a large index, resulting in substantial memory requirements that effectively restricts its use to high-performance computing clusters and smaller databases.

Centrifuge was developed to overcome the memory requirement issue faced by Kraken and enable rapid metagenomic classification on a personal computer. This was accomplished by replacing the k-mer based indexing with a method based on the Burrows-Wheeler transform and Ferragina-Manzini index (Kim et al. 2016). For a database of 4,278 complete prokaryotic genomes, Kraken requires ~100 GB of memory for its index, whereas Centrifuge only requires 4.2 GB. This space-efficient indexing makes it possible for Centrifuge to index the entire NCBI nucleotide (nt) database, which is a comprehensive collection of sequences from viruses, bacteria, archaea, and eukaryotes, making it one of only a few practical alternatives to BLAST for classifying sequences to nt.

A downside to alignment-free methods is that they are more sensitive to third-generation sequencing error profiles. The mismatches and indels prevent exact k-mer matches, increasing the chance of misclassifications, especially when classifying organisms that have high sequence identity. As a result, alignment-free classification methods often have less discriminatory power at lower taxonomic levels compared to inexact alignment strategies such as the one used by BLAST.

1.4.2 Data exploration

The output from classification tools can often be complex and many tools, including BLAST and Centrifuge, give multiple taxonomic assignments per read by default rather than a single LCA assignment, as is the case for Kraken. Tools such as MEGAN (MEtaGenome ANalyzer) and Pavian enable exploration and visualisation of metagenomic classification results.

MEGAN works with output from alignment-based algorithms, such as BLAST, MALT, and DIAMOND, for taxonomic profiling and functional analysis (Huson et al. 2007). The tool uses an algorithm to assign each read to the lowest common ancestor of the set of taxa that the sequence had alignments to. MEGAN provides graphical output, including interactive taxonomic trees, pies, and bar charts, as well as statistical output. The program, written in Java for portability, runs locally on a desktop or laptop, but can require a lot of memory and often suffers from long response times to user inputs. There are two versions of MEGAN: Ultimate edition, which has all of the features and regular updates, but is only available by purchasing a commercial license; and Community edition, which is free to use, but only offers a subset of the features from Ultimate edition and will have no new features added.

In contrast to the heavyweight MEGAN software, Pavian is a lightweight Shiny web app with a backend written in R (Breitwieser and Salzberg 2020). The app is free to use and enables users to visualise and explore results from certain alignment-free classifiers, such as Kraken, Centrifuge and MetaPhlAn. Pavian has a simple interface and features interactive tables, heatmaps and Sankey flow diagrams for exploring and comparing complex metagenomics datasets.

Another web-based tool, One Codex, offers both k-mer based classification and visualisation of results on the same web platform (Minot et al. 2015). Sequencing data is uploaded in FASTA or FASTQ format to the platform via the GUI’s upload tool or a command-line tool. Once uploaded, reads are classified, and an interactive report is generated and linked to the user’s account. Unfortunately, the user has little control over classification parameters, the reference database is not customisable, and after a certain number of free analyses the service must be paid for.

1.4.3 Real-time analysis tools

Regardless of the classification method used, overall time to analysis results for a metagenomic sequencing project still often takes several days as analysis only begins once the sequencing run has finished. This limits the utility of traditional sequencing approaches in settings where rapid species identification is crucial, such as during an outbreak of an infectious disease, or clinical diagnosis of an antibiotic resistant infection. None of the data exploration tools mentioned in the previous section (MEGAN, Pavian, and One Codex) have been specifically designed for nanopore data and they are all incapable of real-time analysis. ONT's own EPI2ME platform is currently the only example of a real-time metagenomic analysis tool.

Two of the most popular workflows on the EPI2ME platform are WIMP (What's In My Pot) and ARMA (antimicrobial resistance mapping application). WIMP classifies reads using Centrifuge with a database of bacterial, viral, and fungal RefSeq genomes and presents a taxonomic tree view of the sample (Juul et al. 2015). ARMA aligns reads with minimap2 against all reference sequences available in the CARD database, the Comprehensive Antibiotic Resistance Database (Alcock et al. 2020), to identify antimicrobial resistance (AMR) genes.

The EPI2ME platform provides near real-time analysis but has limitations due to its closed nature. The workflows offer no flexibility or customisation, including classification parameters and reference database. This makes WIMP unsuitable for analysing some kinds of datasets – e.g. from environments that contain microscopic eukaryotes, such as ocean metagenomes. Additionally, EPI2ME can only be accessed by ONT customers and analyses require credits, known as 'Metricoins', to run. As an online service, the user requires a fast and stable internet connection, and the platform can be prone to lag. Together, these limitations highlight the need for open-source alternatives.

1.5 Nanopore and hybrid approaches for metagenomics

Nanopore sequencing has facilitated the study of complex metagenomic samples by providing long, information-rich, reads in real time using relatively inexpensive and portable hardware. The use of long reads has led to the resolution of complex genomic structures and simplified the task of recovering metagenome-assembled genomes (MAGs) (Cusco et al. 2021; Kinkar et al. 2021; Sereika et al. 2022). Furthermore, the low-cost and portable MinION platform together with the availability of rapid library preparation protocols makes nanopore technology an attractive tool for real-time in-field sequencing of environmental and clinical metagenomic samples. Nevertheless, before embarking on a nanopore metagenomic sequencing experiment a few considerations need to be made. These considerations include target read length, sequencing costs, and availability of protocols and analysis tools.

Target read length will depend greatly on the question being asked and quality of the extracted DNA. If high molecular weight (HMW) DNA can be recovered from the sample and the goal is assembly, long or ultra-long reads (read N50 > 50 kb) can be generated using ONT's Ligation Sequencing Kit or Ultra-Long DNA Sequencing Kit. However, the longer the target read length is, the higher the DNA input requirement becomes for efficient library preparation and sequencing.

For taxonomic characterisation studies, a greater number of shorter reads with a more uniform read length distribution would allow for better species detection and abundance estimates. For greater throughput and uniformity, DNA should be fragmented and size selected prior to library construction with the Ligation Sequencing Kit. Due to the losses that occur during fragmentation and size selection, this approach also has a relatively high DNA input requirement.

If it is not possible to obtain enough DNA from the sample to meet the Ligation kit input requirements, users could consider ONT's PCR-free transposase-based Rapid

Sequencing Kit which has a lower recommended input requirement of 400 ng. However, the DNA still needs to be relatively high molecular weight as the kit is optimised for fragments greater than 30 kb. ONT's Rapid-based kits also achieve lower throughput than the Ligation-based kits.

One of the main advantages of ONT's sequencing platforms is that there is no capital cost for the hardware, only the consumables need to be purchased, making the devices much more accessible for those who wish to do their own sequencing. Sequencing costs per gigabase (Gb) of data range from around £25 to 50 for the MinION and £17 to 35 on their largest scale platform, the PromethION P48. This compares favourably to the £150 per Gb for an Illumina MiSeq Reagent Kit v3 600-cycle run, a popular choice for metagenomics due to the longer 300 bp PE reads. However, 250 bp PE reads can be obtained at a much lower cost, around £15 – 20 per Gb, with Illumina's production scale NovaSeq 6000 platform using an SP flowcell.

To date, most metagenomic studies have relied on high-throughput short-read sequencing. However, these short reads limit taxonomic assignment resolution and result in highly fragmented assemblies. Long nanopore reads can be more taxonomically informative and lead to highly contiguous MAGs (Cusco et al. 2021). Nevertheless, due to higher error rates nanopore-only assemblies still often contain insertions and deletions (indels), especially in homopolymer regions, that can cause frameshift errors during gene calling (Watson and Warr 2019). A widely adopted solution has been to use a hybrid approach, assembling the long nanopore reads and then using short accurate reads for post-assembly error correction (Chen et al. 2021).

Hybrid approaches benefit from the advantages of both read types but suffer from increased costs and complexity. As ONT continue to make improvements to the hardware, software, and chemistry of their platforms, accuracy and yield are likely to improve further, lessening the incentive to use hybrid approaches and resulting in more nanopore-only metagenomic studies (Sereika et al. 2022).

1.6 Thesis scope

This introduction has shown the importance of metagenomics and the growing role that nanopore sequencing is playing in helping to answer questions about microbial communities. Yet tools designed specifically for nanopore analysis are rare and the only user-friendly tool capable of real-time classification is the EPI2ME service provided by ONT, which is not open and has significant limitations.

The overarching aim of this project was to generate analysis methods and tools that facilitate the use of nanopore sequencing technology in the field of metagenomics and for real-time applications. The work presented in this thesis leverages two key features of the technology, long reads and real-time data, and covers the following areas:

- Development of the RevMet (Reverse Metagenomics), a hybrid sequencing approach that utilises long nanopore reads for semi-quantitative characterisation of mixed-species eukaryote samples without the need for complete reference genomes, instead using short-read genome skims (Chapter 2).
- Development of a new software package MARTi, Metagenomic Analysis in Real-Time, an open-source tool that enables real-time analysis and visualisation of metagenomic sequencing data in a lightweight browser interface (Chapter 3).
- Demonstration of MARTi's ability to accurately profile the taxonomic composition of a known microbial mix and carry out real-time characterisation, including anti-microbial resistance gene identification, of clinical samples (Chapter 4).
- A summary of the work presented in this thesis and discussion of future directions (Chapter 5).

2 Chapter 2 – Semi-quantitative characterisation of mixed pollen samples using MinION sequencing and Reverse Metagenomics

This chapter is an adaptation of work that was published in *Methods in Ecology and Evolution*:

Peel, N., Dicks, L.V., Clark, M.D., Heavens, D., Percival-Alwyn, L., Cooper, C., Davies, R.G., Leggett, R.M. and Yu, D.W. (2019). Semi-quantitative characterisation of mixed pollen samples using MinION sequencing and Reverse Metagenomics (RevMet). *Methods in Ecology and Evolution*, 10(10), pp.1690-1701.

2.1 Introduction

Pollination is a key ecosystem service; almost 90% of all flowering plant species, including 75% of food crops (mainly fruits, nuts, and vegetables), rely on animal pollination (Klein et al. 2007; Ollerton et al. 2011). The benefits of pollinators, and pollinator-dependent plants, also include the production of medicines, biofuels, fibres, and construction materials (Potts et al. 2016a). There is growing concern over the decline of wild and domesticated pollinators and the resulting decrease in pollination services and crop production (Potts et al. 2010; Burkle et al. 2013). These declines are thought to be caused by multiple threats acting together, including habitat loss, climate change, and the spread of diseases (Vanbergen et al. 2013).

To mitigate drivers of pollinator decline, the Intergovernmental Science - Policy Platform for Biodiversity and Ecosystem Services (IPBES) has suggested three complementary strategies: (1) ecological intensification, which involves boosting agricultural production by increasing the provision of supporting ecological processes such as biotic pest regulation, nutrient cycling, and pollination (Bommarco et al. 2013; Tittonell 2014); (2) strengthening existing diversified farming systems, including gardens and agroforestry, for the generation of ecosystem functions; and (3) investment in ecological infrastructure, to protect, restore, and connect natural and semi-natural habitats across agricultural landscapes, so that pollinator species can more easily disperse and find nesting and floral resources (Potts et al. 2016b).

However, knowledge gaps limit the effectiveness of these strategies (Dicks et al. 2013; Wood et al. 2015). For instance, it is still not clear which plant species are the most valuable food resources and how plant species vary in value across pollinator species, over time, and in different environmental conditions. It is also not well understood whether the addition of floral resources might draw pollinators away from pollinator-dependent crop plants (Morandin and Kremen 2013), or whether floral enhancement will alter levels of plant-target specialism, at the levels of insect species and of individual insects, resulting in changes in pollination efficiency (Morales and Traveset 2008; Lucas et al. 2018).

Therefore, a crucial technical challenge for understanding plant-pollinator interactions is to develop a method to identify and quantify the species of pollen that are consumed by pollinators. Identifying and quantifying pollen has traditionally been carried out by using light microscopy to distinguish plant species by grain morphology, a labour-intensive technique that requires expert knowledge and lacks discriminatory power at lower taxonomic levels (Khansari et al. 2012; Long and Krupke 2016). In contrast, high-throughput DNA sequencing now allows pollen identification without expert knowledge of pollen morphology and taxonomy.

The currently dominant sequence-based method is metabarcoding, which involves amplifying taxonomically informative marker genes from mixed samples via polymerase chain reaction (PCR) (Ji et al. 2013). The resulting amplification products, or amplicons, are sequenced, and the reads are assigned to taxonomies by matching against barcode databases, such as the Barcode of Life Data System (Ratnasingham and Hebert 2007). Notably for plants, there is no single barcode gene that matches the resolving power of 16S rRNA for prokaryotes and Cytochrome Oxidase (CO1) for animals (Hollingsworth et al. 2016). Instead, plant-related barcoding studies rely on a combination of marker genes, which include plastid regions *rbcL* and *matK* and the internal transcribed spacer (ITS) regions of nuclear ribosomal DNA (Li et al. 2015; Hollingsworth et al. 2016). Metabarcoding of mixed-species pollen samples can reveal the presence and absence of constituent plant species (or genera), but there are strong reasons to doubt that metabarcoding can accurately quantify their relative abundances, due to PCR amplification biases and varying copy numbers of barcode loci (Keller et al. 2015; Richardson et al. 2015; Sickel et al. 2015; Bell et al. 2017; Lamb et al. 2019).

In contrast to the targeted-sequencing approach of metabarcoding, ‘shotgun metagenomics’ involves randomly sequencing short stretches of genomic DNA from mixed samples. In standard metagenomics, these short reads (‘queries’) are mapped to either assembled genomes or to collections of barcode genes (‘references’), which creates a requirement for large numbers of reference genomes (Sharpton 2014) or barcodes (Zhou et al. 2013), with the latter being very inefficient (Ji et al. 2020). Species identification is obtained by first calculating a similarity metric between each short read and each reference sequence (e.g., % identity) and then using an algorithm to assign each short read to the most likely reference sequence. The potential key advantages of shotgun metagenomics are that it can avoid the PCR-induced biases seen with metabarcoding, especially if PCR-free library preparation protocols are used (Jones et al. 2015; Nayfach and Pollard 2016) and that by sampling across the whole genome, variation in the copy numbers of a few loci is rendered less important. However, the requirement for reference genomes means that most shotgun metagenomics studies focus on prokaryotic organisms, since large numbers of prokaryote reference genomes are available. In contrast, eukaryotes are not well represented in sequence databases due to being more expensive to sequence and assemble (Gilbert and Dupont 2011). As a result, eukaryotes have mostly been neglected in metagenomic studies (Escobar-Zepeda et al. 2015).

In this chapter, a new metagenomic pipeline for eukaryotes is described. The method avoids the need for assembled reference genomes, instead, each reference species is represented by a 'genome skim' (Straub et al. 2012), which is a low-cost, low-coverage, shotgun dataset, i.e. simply a set of short reads. The sets of short reads are used to classify individual long reads from pollen that have been generated by sequencing mixed-species pollen loads with the MinION. The pollen reads need to be long reads, so that multiple reference reads map to them, this providing classification accuracy.

To demonstrate the effectiveness of the method, reference genome skims were generated for 49 wild UK plant species and used to identify and quantify plant species in two kinds of query samples: mock plant DNA mixtures of known composition and mixed-species pollen samples collected from wild bees. Each of the long reads in the query samples are individually classified and the proportion of long reads assigned to a plant species is shown to be a reasonably accurate estimate of that species' frequency in a mixed-species sample, based on relative quantities of DNA. The pipeline is called Reverse Metagenomics, or RevMet, because reference sequences are mapped to query sequences, which is the reverse of the standard metagenomic approach of mapping reads to (assembled) references.

All experimental work presented in this chapter was performed by the author, except for the in-field sample collections, which were carried out by Lynn Dicks, Richard Davies, and Chris Cooper. Lawrence Percival-Alwyn provided training and assistance during the initial CTAB-based phenol-chloroform extractions and Darren Heavens provided support and guidance for the first nanopore runs.

2.2 Methods

2.2.1 Sampling of bees and plant tissue

Sample collection was carried out in the Pensthorpe Natural Park area (52°49'23"N, 0°53'14"E) of Norfolk, UK, over the course of four days during June and July 2016. Leaf samples were collected from all plant species with open flowers, including grasses and trees, within a 100 m radius of the collection site (n = 49 species). The 100 m radius was chosen to capture the likely area of flowering plants covered by an individual bee in a pollen-foraging bout. Bees actively collecting pollen are assumed to be on ‘exploitation flights’, defined for bumblebees by Woodgate et al. (2016) as single loop flights to a previously known location for the sole purpose of foraging, rather than ‘exploration flights’, which cover a much larger area. In the data reported by Woodgate et al. (2016), foraging activity on *Bombus terrestris* exploitation flights was usually constrained within a circle of radius 100m.

Leaf tissue was preserved on dry ice in the field followed by storage at -80 °C. Foraging wild bees (n = 48: 9 *Apis mellifera*, 27 *Bombus terrestris/lucorum complex*, 12 *Bombus lapidarius*) were collected with hand nets or into falcon tubes directly from flowers and euthanized in falcon tubes containing ethanol-soaked tissue paper. Pollen loads were scraped from bee corbiculae using a mounted needle and stored in absolute ethanol. The plant species on which each bee was foraging when collected was recorded. Leaf tissue DNA extraction, library preparation, and Illumina sequencing

2.2.2 Leaf tissue DNA extraction, library preparation, and Illumina sequencing

Leaf tissue from each of the 49 plant species was disrupted by bead-beating using a 4-mm stainless steel bead with a Qiagen TissueLyser II running at 22.5 Hz for 4 min, rotating the adapter sets after 2 min. DNA was extracted using the DNeasy Plant Kit (Qiagen, Hilden, Germany) following manufacturer's instructions. DNA concentrations were measured on a Qubit 2.0 fluorometer (Thermo Fisher, Waltham, USA) using the dsDNA HS assay kit, and fragment size distribution was checked with a Genomic DNA Analysis ScreenTape on the TapeStation 2200 (Agilent, Santa Clara, USA).

The Earlham Institute (Norwich, UK) applied a modified version of Illumina’s Nextera protocol, known as Low Input Transposase Enabled (LITE) protocol (Perez-Sepulveda et al. 2020), to generate a separate sequencing library for each leaf sample, targeting an average insert size of 500 bp. The LITE libraries were then pooled based on estimated genome sizes (Appendix 1), obtained from the Royal Botanic Gardens Kew Plant DNA C-values database (Pellicer and Leitch 2020), in order to achieve 0.5x coverage of each species genome. The pooled libraries were sequenced on one lane of Illumina HiSeq 2500 in Rapid Run mode (250 bp PE).

2.2.3 Construction and sequencing of mock pollen samples

DNA from twelve of the 49 plant species were used to construct six mock communities. Each mock was made using 200 ng DNA in total, with species added at different proportions: 0.08% to 45.25% (Table 2.1). For each mock, technical-replicate pairs were prepared using ONT’s (Oxford, UK) Rapid Barcoding Sequencing Kit (SQK-RBK001), following the RBK_9031_v2_revl_09Mar2017 version of the manufacturer’s protocol. The 12 libraries (six mocks, duplicated) were sequenced on a single MinION R9.5 flow cell (FLO-MIN107).

Table 2.1 DNA mock community compositions

	Knautia arvensis	*Galium verum*	*Crepis capillaris*	*Papaver somniferum*	*Anagallis arvensis*	*Sambucus nigra*	*Bryonia dioica*	*Ranunculus repens*	*Lotus corniculatus*	*Digitalis purpurea*	*Leucanthemum vulgare*	*Stachys sylvatica*
MM1.Ratios	100	100	100	10	10	10	1	1	1	0	0	0
MM2.Ratios	0	0	0	1	1	1	10	10	10	100	100	100
MM3.Ratios	0	0	0	0	0	0	10	100	0	0	0	0
MM4.Ratios	0	0	0	1	0	1000	0	0	100	0	0	100
MM5.Ratios	100	100	100	0	1	1	1	100	0	0	0	1
MM6.Ratios	1	1	1	100	100	100	100	0	0	0	0	1
MM1.DNA (ng)	60.1	60.1	60.1	6.0	6.0	6.0	0.6	0.6	0.6	0.0	0.0	0.0
MM2.DNA (ng)	0.0	0.0	0.0	0.6	0.6	0.6	6.0	6.0	6.0	60.1	60.1	60.1
MM3.DNA (ng)	0.0	0.0	0.0	0.0	0.0	0.0	9.1	90.5	0.0	9.1	90.5	0.9
MM4.DNA (ng)	0.0	0.0	0.0	0.2	0.0	166.5	0.0	0.0	16.7	0.0	0.0	16.7
MM5.DNA (ng)	49.5	49.5	49.5	0.0	0.5	0.5	0.5	49.5	0.0	0.0	0.0	0.5
MM6.DNA (ng)	0.5	0.5	0.5	49.5	49.5	49.5	49.5	0.0	0.0	0.0	0.0	0.5
MM1.Percentages	30.0%	30.0%	30.0%	3.0%	3.0%	3.0%	0.3%	0.3%	0.3%	0.0%	0.0%	0.0%
MM2.Percentages	0.0%	0.0%	0.0%	0.3%	0.3%	0.3%	3.0%	3.0%	3.0%	30.0%	30.0%	30.0%
MM3.Percentages	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	4.5%	45.3%	0.0%	4.5%	45.3%	0.5%
MM4.Percentages	0.0%	0.0%	0.0%	0.1%	0.0%	83.3%	0.0%	0.0%	8.3%	0.0%	0.0%	8.3%
MM5.Percentages	24.8%	24.8%	24.8%	0.0%	0.3%	0.3%	0.3%	24.8%	0.0%	0.0%	0.0%	0.3%
MM6.Percentages	0.3%	0.3%	0.3%	24.8%	24.8%	24.8%	24.8%	0.0%	0.0%	0.0%	0.0%	0.3%

2.2.4 Bee-collected pollen DNA extraction, library preparation, and MinION sequencing

After removing storage ethanol from the 48 bee-collected pollen loads, the pollen was disrupted with ca. five 1-mm stainless steel beads for 2 min at 22.5 Hz using a Qiagen TissueLyser II, rotating the adapter sets after 1 min. The pollen samples were resuspended in 600 µl CTAB extraction buffer (2% CTAB, 1.4 M NaCl, 20 mM EDTA, pH 8.0, 100 mM Tris-HCl pH 8.0), 0.5 µl of β-Mercaptoethanol, 4 µl of proteinase K, and vortexed for 5 s. Following a 1 hr incubation at 55 °C, the tubes were centrifuged for 6 min at 18,000 x g. The ≈500 µl of supernatant was extracted to a clean 1.5 ml tube before an equal volume of chilled (2-8 °C) Phenol:Chloroform:Isoamyl Alcohol (25:24:1, v/v) was added to the lysate. The samples were vortexed for 10 s (5 x 2 s bursts), centrifuged for 5 min at 14,000 x g, and the upper aqueous phase (≈ 420 µl) was extracted by pipette and transferred into a clean 1.5 ml tube.

An equal volume of Agencourt AMPure XP beads was added to each sample, vortexed for 20 s (10 x 2 s bursts), and then incubated for 10 min at room temperature. The beads were separated solution by placing the samples onto a magnetic tube rack for 5 min, and the cleared supernatant was removed by aspiration. The beads were washed twice using the following protocol: 1 ml of 80% ethanol was added, incubated at room temperature for 30 s, and then removed, followed by air drying for ≈ 3 min. The magnetic beads were resuspended in 55 µl of EB (Elution Buffer: 10 mM Tris-HCl) and incubated at 37 °C for 10 min. The tubes were placed back onto the magnetic rack to bind the beads, and the eluted DNA (≈ 50 µl) was transferred into fresh tubes. A 1 µl aliquot of 1-in-10 diluted Qiagen RNase A was added to each DNA sample before being incubated for 30 min at 37 °C. The concentration of the eluted DNA was assessed using the dsDNA HS assay on a Qubit 2.0 fluorometer. To check the DNA for degradation, fragment size distributions were checked with a TapeStation 2200 using the Genomic DNA Analysis ScreenTape.

Finally, the extracted DNA was prepared and sequenced using the same protocol as used for the DNA mocks above, except that only one library was prepared for each sample. At the time of the work, only twelve samples could be multiplexed using the Rapid Barcoding Sequencing Kit (now 96 can be multiplexed with the Native Barcoding Expansion); thus, four flow cells were required. Due to continuous software upgrades by ONT, the specific software versions of MinKNOW varied across runs and is recorded in the final sequence files (fast5 format), which are available from the EBI's European Nucleotide Archive (see Data availability).

2.2.5 Illumina and MinION read pre-processing

Duplicate reads were removed from the 49 plant-reference Illumina datasets using *NextClip 1.3.2* (Leggett et al. 2014), and then *cutadapt 1.10* (Martin 2011) was used to trim Illumina adaptors and filter out reads shorter than 100 bp. The resulting unmerged FASTQ files constitute the 49 reference skims.

The MinION datasets from the 12 mocks and the 48 pollen loads were basecalled and demultiplexed with *albacore 2.1.10* (ONT). The resulting FASTQ files were converted to FASTA format. Long reads deriving from plant organelles were removed because they are highly conserved across plant species and in pilot tests mapping to organellar long reads resulted in a higher rate of incorrect assignments than mapping to nuclear long reads (data not shown). NCBI Entrez (https://www.ncbi.nlm.nih.gov/sites/batchentrez) was used to download 2,583 Land Plant organelle genomes, 2,357 plastid and 226 mitochondrial. Organelle reads were identified by aligning each of the MinION datasets to the organellar genomes using *minimap2 2.7* (Li 2018) and removed from the FASTA files. The resulting 60 (= 12 + 48) organelle-filtered FASTA files constitute the mock and pollen query datasets, and in the next step, the 49 plant reference skims were used to assign a taxonomy to each long read in the mock and pollen query datasets (Figure 2.1c).

2.2.6 Taxonomic assignment of mock-sample and bee-collected pollen MinION reads

Illumina reads from each of the 49 reference skims were mapped against every individual long read in each of the mock and bee-collected pollen datasets using *bwa mem 0.7.17* (Li 2013). *SAMtools 1.7* (Li et al. 2009) was used to remove unmapped reads and secondary and supplementary alignments. After SAMtools indexing, the depth of mapping coverage at each long-read position was calculated using the SAMtools depth function. A python script, *percent_coverage_from_depth_file.py*, was used to calculate the 'percent coverage' for each long read - defined as the fraction of nucleotide positions that were mapped to by one or more reference-skim Illumina reads.

Each long read was assigned to the plant species that mapped with the highest percentage coverage, unless the highest percent coverage was <15%, in which case the long read's identity was judged ambiguous and left unassigned. Additionally, for clarity of presentation, a 1% minimum-abundance filter was implemented, removing plant species represented by fewer than 1% of the total assigned long reads in each sample.

All of the bioinformatic steps for taxonomic assignment can be run on a laptop/desktop computer, but for this study the pipeline was run in parallel on a High-Performance Computing (HPC) cluster for greater performance.

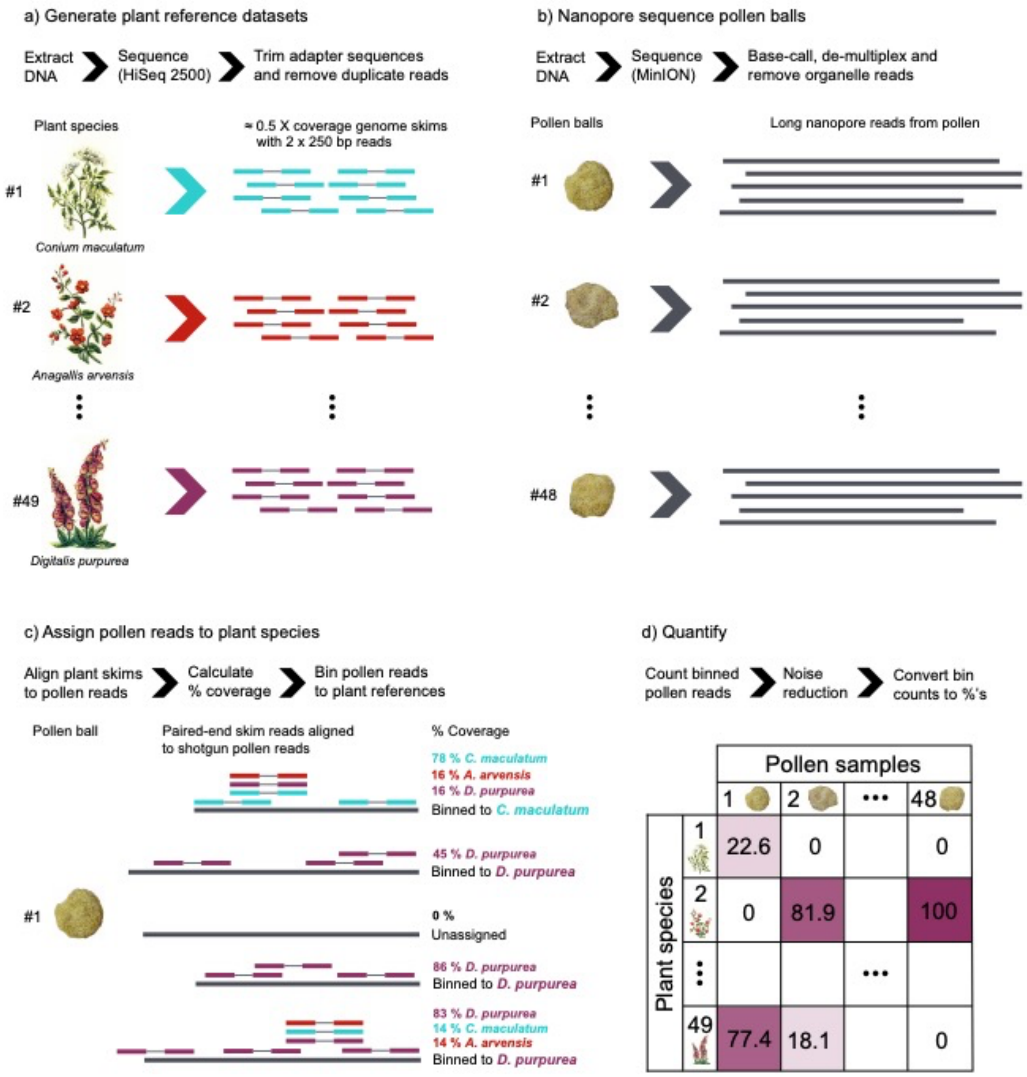
2.3 Results

2.3.1 A reference set of plant genome skims

Low genome coverage, short-read, shotgun-sequencing datasets (‘reference skims’) were successfully generated for all 49 plant species (Figure 2.1a). After pre-processing, the mean estimated coverage was 0.6x (0.1 to 1x, details in Appendix 1).

2.3.2 Mock DNA mixes

The six mock communities, each with two technical replicates, were sequenced on a MinION. These produced relatively short reads for nanopore sequencing, with mean length 1,914 bp (longest 41,058 bp), likely due to the low quantity and molecular weight of the input DNA (discussed later). After demultiplexing, 88.8% of the reads could be assigned to one of the 12 mock mixes, with the remaining reads left unclassified. Sequences originating from organellar genomes made up between 5.1% (MM4.2) to 10.2% (MM3.2) of the reads in the mocks and were removed. The remaining number of reads per mock ranged from 733 (MM2.1) to 2,174 (MM4.1), mean 1,347.



	1	2	•••	48
1	22.6	0		0
2	0	81.9		100
⋮			•••	
49	77.4	18.1		0

Figure 2.1 RevMet pipeline overview. (a) Low coverage, short-read, reference datasets were generated for 49 wild plant species. (b) Bee-collected pollen loads were sequenced on a MinION, generating long read datasets. (c) The 49 short-read reference datasets were separately mapped to the long-read pollen datasets, and each pollen read was assigned to the plant species that mapped with the highest percent coverage or left unassigned if the highest coverage was <15%. (d) Binned pollen reads were counted, noise was reduced by implementing a 1% minimum-abundance filter, and then the remaining bin counts were converted to percentages.

2.3.3 Taxonomic assignment of mock-sample MinION reads

The 49 reference skims were separately mapped to each long read in each of the 12 mock mixes, and each long read was assigned to the plant species that mapped with the highest percent coverage, or left unassigned if the highest coverage was <15%. In total, 65.5% of the mock reads were assigned to a plant species, with 94.7% of those reads being assigned to a species known to be present in that mock sample. Almost all (93.4%) of the 563 false-positive read assignments were made to one species, *Ranunculus acris*, and all these assignments occurred in the mock samples that contained the very closely related species *Ranunculus repens*. The few other false-positive assignments all occurred at a rate of less than 1% of the assigned long reads in their mixes and for presentational clarity are not shown in Figure 2.2. The full results are in Appendix 2.

All of the plant species that had been added to the mock compositions at proportions ≥1% were detected by RevMet in at least one of the two replicates, and in all cases the frequencies of long reads assigned to each plant species were reliably 'semi-quantitative' and could differentiate low- and high-abundance plant species (Figure 2.2). In general, the technical replicates showed a high level of repeatability, although in two of the mocks there was one species in each (*Lotus corniculatus* in MM2 and *Digitalis purpurea* in MM3) that was detected in only one of the two replicates. In two of the mocks, there was one species each (*Lotus corniculatus* in MM2 and *Digitalis purpurea* in MM3) that were detected in only one of the two replicates. In these two replicates, *Lotus corniculatus* and *Digitalis purpurea* were expected to be present at only 3.0% and 4.6%, respectively. Both species were consistently underrepresented across the mock data sets, which suggests that the DNA quantification may have been inaccurate prior to the creation of the mocks.

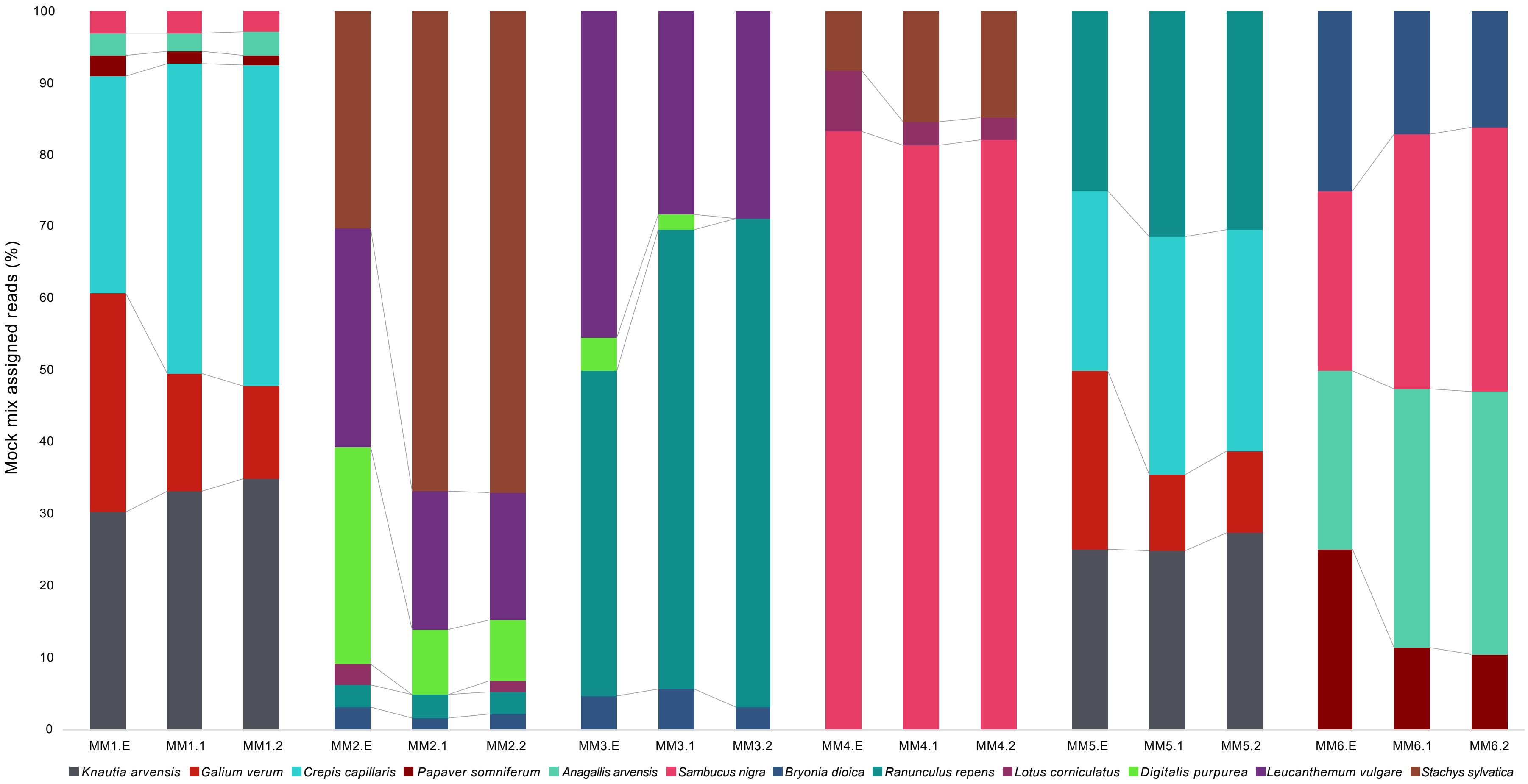


Figure 2.2 Expected versus observed mock mix compositions. Six mock plant DNA mixes, each with two technical replicates, were sequenced on a MinION and the RevMet method was applied. The first stacked bar of each triplet represents the expected proportions based on input DNA. The second and third bar of each triplet reflect the observed MinION read assignments resulting from this pipeline

2.3.4 Reference-skim subsampling

To estimate a minimum recommended depth of coverage needed per reference skim, the *Knautia arvensis* genome skim, which is a major constituent species in mock mixes MM1 and MM2, was subsampled. The skim was randomly subsampled from its maximum of 0.65x down to 0.05x, in steps of 0.05x using a custom script. For each subsample, the whole pipeline was re-run along with the full reference skims of the other 48 plant species. The number of mock reads assigned to *Knautia arvensis*, and the number of unassigned reads, at each level of coverage was recorded. This subsampling was repeated three times (Figure 2.3).

As expected, the larger the reference-skim dataset size for *Knautia arvensis*, the more reads in the MM1 and MM2 mocks were assigned to this species and the fewer reads left unassigned. Importantly, the rate of increase was decelerating; over half of the MinION reads that were assigned to *Knautia arvensis* with a 0.65x genome skim could also be assigned with just a 0.1x skim, even though all the other reference skims in the mapping run were kept at their original sizes.

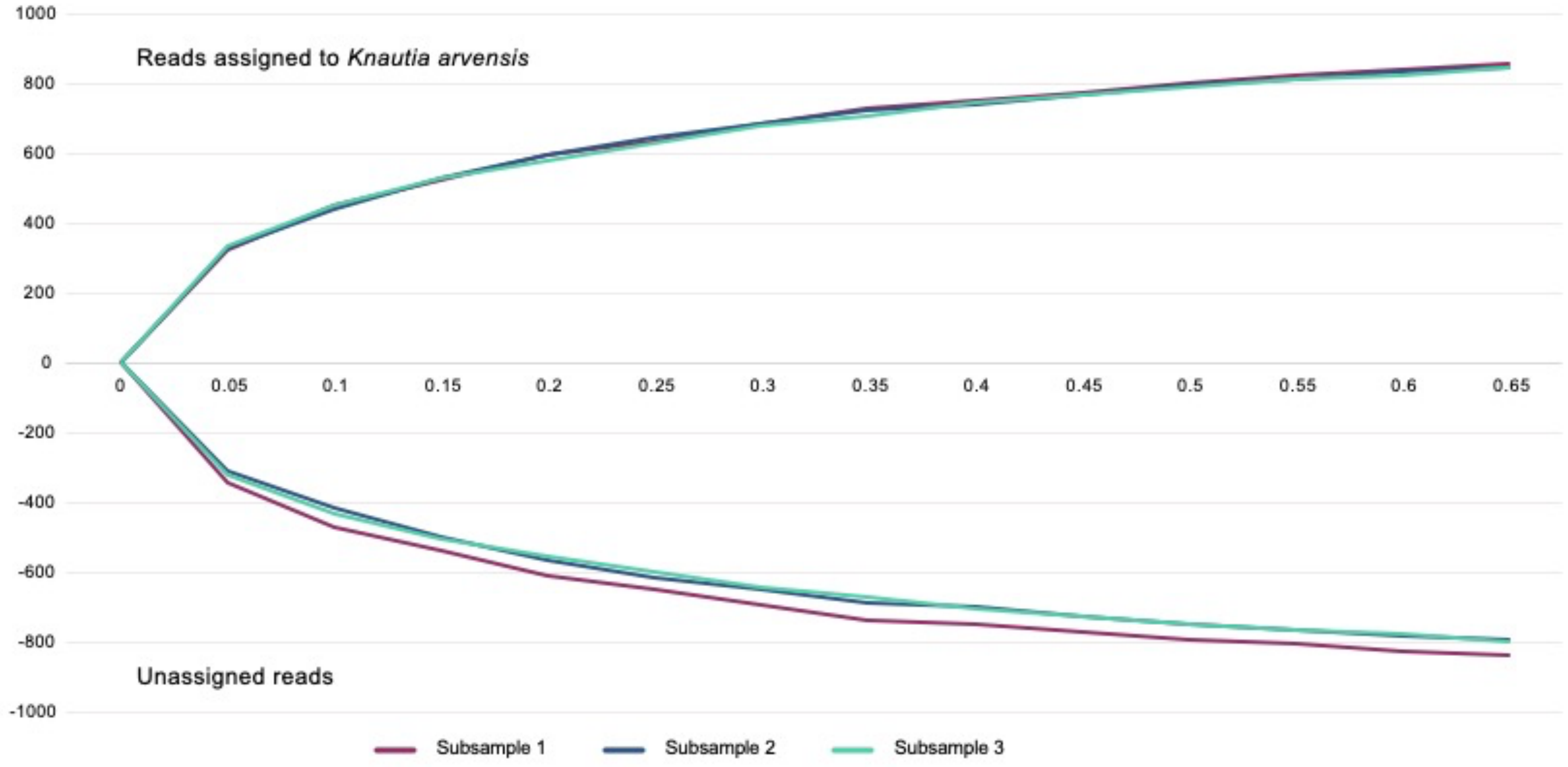


Figure 2.3 Reference skim subsampling. The Knautia arvensis genome skim, which is a major constituent species in mock mixes MM1 and MM2, was randomly subsampled from its maximum of 0.65x down to 0.05x, in steps of 0.05x. For each subsample, the whole RevMet pipeline was re-run along with the full reference skims of the other 48 plant species. The number of mock reads assigned to Knautia arvensis, and the number of unassigned reads, at each level of coverage was recorded. This subsampling was repeated three times. The decrease in number of unassigned reads was roughly equal to the increase in assigned Knautia arvensis reads at each genome skim size.

2.3.5 Taxonomic assignment of bee-collected pollen MinION reads

The 48 bee-collected pollen loads harvested from the corbiculae of three species, *Apis mellifera*, *Bombus terrestris/lucorum complex*, and *Bombus lapidarius*, yielded DNA quantities ranging from 191 to 3,750 ng, and all successfully produced libraries, demonstrating that pollen carried by individual bees can provide sufficient DNA for MinION sequencing. After demultiplexing, the mean read number per pollen sample was 2,430, with an average length of 2,300 bp (longest 51,629 bp).

As with the 12 mocks, each of the reference skims was aligned to each long read in each of the 48 pollen samples, the long reads were either assigned to the plant species achieving the highest percent coverage or left unassigned, and any plant species assigned fewer than 1% of the long reads in each bee-collected pollen sample was filtered out. In total, 49.7% of the long reads were assigned to one of the reference plant species. In 38 of the 48 bees (79.2%), pollen from the plant species on which each bee was captured was found to be present in that bee's pollen load (Appendix 4).

Each of the 48 pollen loads was found to contain one or two major plant species (defined as read frequency ≥10%) (Figure 2.4a). All nine of the *Apis mellifera* pollen loads contained a single major species, whereas 16 of 27 *Bombus terrestris/lucorum complex* and 6 of 12 *Bombus lapidarius* pollen loads were comprised of two major species (Figure 2.4a). These differences in mean number of major species were statistically significant (*Apis mellifera* versus *Bombus terrestris/lucorum complex* (Welch's t-test, t = -6.15, df = 26, p-value < 0.0001) and versus *Bombus lapidarius* (t = -3.32, df = 11, p-value < 0.01)) (Figure 2.4b). Another way of visualising the wild-bee results is as a plant-pollinator network graph (Figure 2.4c). The diagram was constructed for the 48 wild-bee pollen samples, using the bipartite 2.11 package (Dormann et al. 2009) for the R statistical language (R Core Team 2018). For presentational clarity, only plant species represented by more than 10% of the assigned reads in each sample are shown. Overall, 6 of the 49 reference plant species were identified as major components in the 48 pollen loads, and the majority of bee-collected pollen samples were dominated by one plant species.

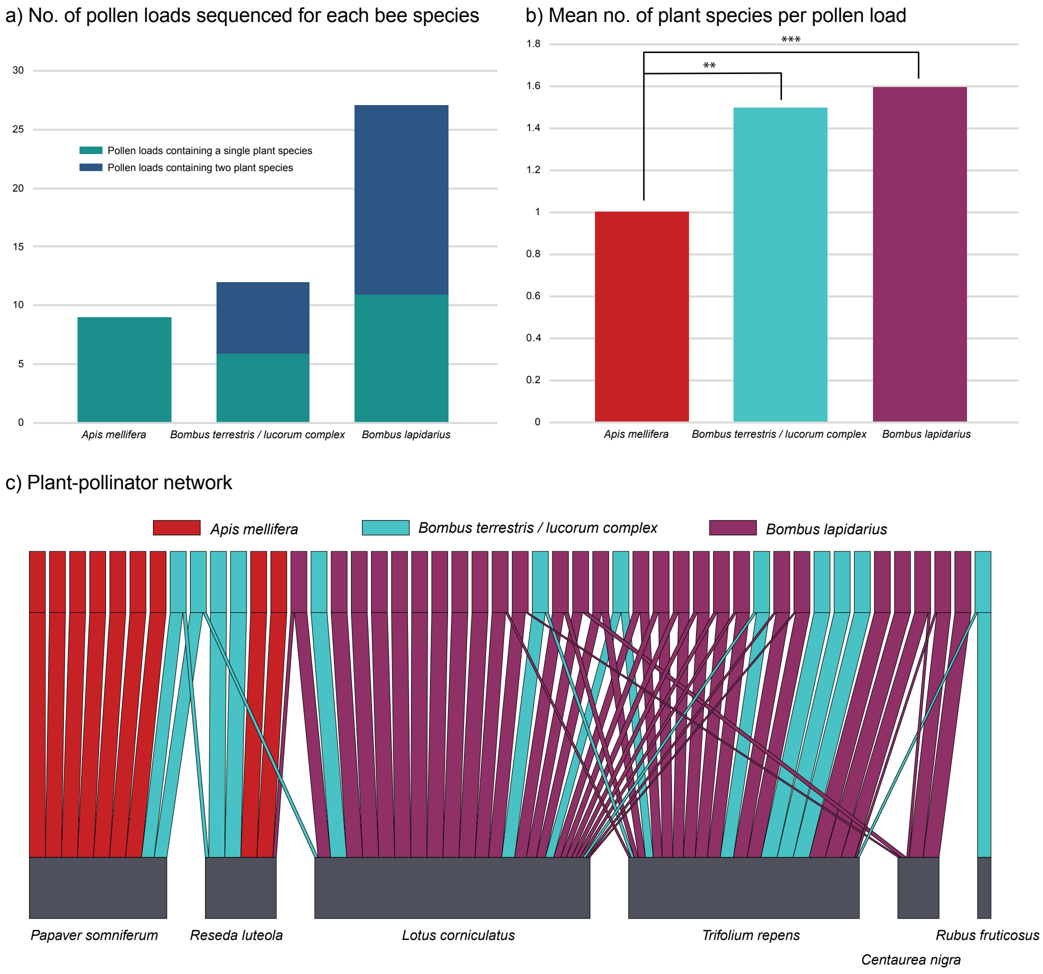


*Figure 2.4 Bee-collected pollen compositions and plant–pollinator interactions. (a) The number of individual pollen loads sequenced from three different species of bee. The proportion of pollen loads that contained a single major plant species are represented by green bars, while those with two major plant species are shown in blue. (b) Mean number of plant species per pollen load for each of three different species of bee: **p < .01, ***p < .001. (c) Bipartite plant–pollinator network. The upper bars represent individual pollen loads from three different bee species, Apis mellifera (red), Bombus terrestris/lucorum complex (blue), and Bombus lapidarius (purple). The lower bars (grey) represent plant species. Link width indicates the MinION read proportion of each major plant species within each pollen load.*

2.4 Discussion

Using light microscopy to identify plant species from pollen requires expert knowledge and is costly when applied to many samples (Khansari et al. 2012). There is a need for a quick and low-cost method that can be scaled to large numbers of pollen samples. Metabarcoding is the current leading candidate, but there are concerns over its discriminatory power at lower taxonomic levels, and there is good reason to believe that metabarcoding is not reliably quantitative (Keller et al. 2015; Richardson et al. 2015; Sickel et al. 2015; Bell et al. 2017; Lamb et al. 2019). A PCR-free shotgun-metagenomics approach has greater potential for providing reliable quantitative analysis with high power for resolving species. However, applying shotgun metagenomics to eukaryotes is challenging due to the lack of reference genomes (Gilbert and Dupont 2011). We have developed a metagenomics method that avoids the need for reference genomes. Instead, each reference species is represented by just a low-cost genome skim, and we use a set of such skims to identify individual long reads from pollen samples, produced by the MinION sequencer.

We evaluated the RevMet pipeline with mock DNA mixtures of known composition and then applied the pipeline to pollen collected from wild bees. The main findings are:

1. RevMet can identify plant species present in mixed-species samples at proportions of DNA ≥1%, with few false positives and false negatives, and can reliably differentiate species represented by high versus low amounts of DNA in a sample (Figure 2.2, Appendix 2).
2. Genome skims with sequence coverage as low as 0.05x can be used for detecting species presence and for estimating relative abundance in terms of DNA mass. Increasing skim coverage increases detection power, at a decelerating rate (Figure 2.3).

3. Individual pollen loads collected from wild *Apis* and *Bombus* bees yield enough DNA for MinION sequencing (Appendix 3) and generate plausible plant-pollinator networks, as evidenced by the fact that (a) 56.3% of the plant species on which the bees were collected were also the dominant constituent of the corresponding pollen sample (and 79% of plant species on which the bees were collected were detected in the corresponding pollen sample) (Appendix 4), and (b) pollen species richness and compositions were more similar within bee species than across bee species (Figure 2.4).
4. The per-plant-species cost of a reference skim was £90, and the per-pollen-sample cost was £61, including DNA extraction, library preparation, and sequencing. Sequencing costs will likely drop further, given the new Illumina NovaSeq and increases in nanopore yield.

2.4.1 Semi-quantitative species compositions

Roughly 65% of the mock-mix MinION reads and just under 50% of the pollen-load MinION reads could be assigned to the reference plant species. There are several ways that the fraction of assigned nanopore reads could be increased without compromising the assigned proportions or increasing the false positive rate. First, higher depth reference skims could be used. This increases the chance of overlap occurring between the long nanopore query reads and the short-read reference skims.

The average length of the bee pollen reads was 2,300 bp. Generating longer nanopore query reads would increase the chance of overlap with Illumina skim reads and decrease the proportion of unassigned reads. At 2,300 bp, nanopore reads have almost identical recall to the longest Illumina reads (300 bp) for plants in traditional metagenomics (Pearman et al. 2020). However, using Illumina reads for both the pollen and reference skims with the RevMet approach would have led to more false positives and a lower proportion of reads being assigned. If we had complete assembled plant reference genomes, Illumina pollen reads may have outperformed our nanopore read set in a traditional metagenomics study.

During mapping, each Illumina read only maps to one nanopore read within a nanopore barcode dataset. Therefore, the larger the nanopore dataset, the lower the fraction of assigned reads becomes as the long reads “mop up” the Illumina reads. Subsampling nanopore datasets down increases the fraction of assigned reads (data not shown) and decreases analysis time. Alternatively, the nanopore read files could be split into individual reads and all of the skim reads could be mapped to each individual nanopore read. This would increase the fraction of assigned reads but also greatly increase analysis time.

Importantly, the frequencies of nanopore reads that were assigned to each reference plant species were reliably ‘semi-quantitative’, that is, able to differentiate low- and high-frequency plant species, based on DNA mass (Figure 2.2). Within low- and high-abundance categories, accuracy was lower. For example, in mock sample MM1, *Knautia arvensis*, *Galium verum*, and *Crepis capillaris* were the three high-abundance species (each representing 30.3% of total input DNA mass each), and *Papaver somniferum*, *Anagallis arvensis*, and *Sambucus nigra* were the three low-abundance species (each representing 3.0% of total input DNA mass each). The RevMet pipeline estimated the three high-abundance frequencies at means of 34.0%, 14.7%, and 44.0%, and the three low-abundance species at 1.4%, 3.0%, and 3.0%, respectively (Appendix 2).

There are at least three reasons for the remaining quantitative error. First, although 0.5x per reference skim was targeted, coverage still varied across species (Appendix 1), resulting in different powers of discrimination, as shown by the experiment with subsamples of *Knautia arvensis* (Figure 2.3). Fortunately, this study found that even very low-depth skims of 0.05x are useful for species detection and are probably still useful for differentiating rare from abundant species (albeit with more error) (Figure 2.3). Genome sizes are also estimated with error, so it is also helpful that the subsampling experiment suggests that detection power asymptotes with higher sequencing depth (Figure 2.3), and as sequencing costs fall further, a more robust protocol could be achieved by targeting 1x coverage. In cases where coverage of reference skims varies greatly, they should be subsampled down to a uniform level.

Second, very closely related species can generate false positives. The reference-skim database included six congener pairs, and two of the pairs were included (*Papaver* and *Ranunculus*) in the mock mixes. In the case of *Papaver*, there were no *P. rhoeas* false-positives greater than the 1% minimum-abundance filter in the mocks that contained *P. somniferum* (MM1 and MM6) (Appendix 2). In contrast, *Ranunculus acris* was regularly incorrectly assigned to reads in mock mixes that contained the closely related congener *Ranunculus repens*. In fact, almost all the false-positive assignments (93.4%) were to *R. acris*. In retrospect, this result is expected because these two species are not easily differentiated by pollen morphology (Forup and Memmott 2005), floral morphology, or even DNA barcodes (rbcL (99.1% similarity), matK (96.9%), ITS2 (95.5%)).

Third, at the time of the experiments, MinION reads had relatively high error median rates of roughly 5 to 8% depending on the flow cell and kit used (Leggett and Clark 2017). Although this is dropping over time (typically < 5% now), this error rate unavoidably obscures differences between species (although not enough to confound the two *Papaver* species). One of the advantages of the RevMet approach is that it uses percent coverage as a predictor of species presence (Figure 2.1c). Using mapped read counts alone, several instances of low numbers of long reads being given false-positive assignments were observed (data not shown). The percent-coverage filter requires many reference-skim reads to independently identify a species before an assignment is made.

2.4.2 Reference skims

The RevMet pipeline is relatively low cost. In this study, skims for 49 plant species were generated, with genome sizes ranging from roughly 290 Mb (*Epilobium hirsutum*) to just under 15 Gb (*Sambucus nigra*), targeting 0.5x coverage. All skims were produced on a single lane of Illumina HiSeq 2500 in Rapid Run mode (250 bp PE) at a mean coverage of 0.57x. The average cost per skim in this study was just under £90, which includes the DNA extraction, LITE library preparation, sequencing, and data QC. The per skim cost will be lower in studies with smaller eukaryotic genomes, and with the use of Illumina's newer sequencer, the NovaSeq 6000, equivalent skims would cost ~£30 (250 PE with the SP flow cell). Genome-assembly campaigns are also likely to produce more skim datasets for free download in the future.

It is likely that the reference-skim database missed some bee forage plants that were flowering within the foraging ranges of the focal bee species, which greatly exceed 100 m (Dicks et al. 2015). For studies considering pollen collected at colony or nest level, rather than individual level, collecting all the flowering species within a radius of at least 1 km would be advisable, to increase the chance of covering all potential bee forage species within foraging range. As availability of skim databases increases, this will be a less onerous task.

2.4.3 MinION sequencing

ONT's first iteration of the Rapid Barcoding Kit (RBK001) was used to prepare the nanopore sequencing libraries. This kit relies on transposase to randomly fragment DNA and simultaneously add barcoded adapters. Longer read lengths have an increased likelihood of accurate species assignment because they carry more sequence information. The two main ways to obtain longer reads with transposase-based preparations are to: (1) increase the ratio of DNA to transposase e.g., by increasing the input material or by heat killing a proportion of the transposase (which also lowers sequencing yields); and (2) use higher molecular weight input DNA. Since the release of RBK001, ONT's chemistry has evolved, and their Rapid-based kits have seen greater sequencing yields. However, the recommended input for the latest iteration of the Rapid Barcoding Kit (RBK004) is now higher, 400 ng of DNA per sample. That said, input biomasses similar to those used in this study, 200 ng, should still be adequate. Also, even 400 ng is achievable, as 36 of 48 of the wild-bee pollen samples yielded >400 ng (Appendix 3).

As throughput has increased since this study was carried out (a single MinION flow cell can now produce up to 30 gigabases of sequencing data), ONT have released the Native Barcoding Expansion 96 kit (EXP-NBD196) to enable PCR-free multiplexing of up to 96 samples. This expansion works in conjunction with the Ligation Sequencing Kit (SQK-LSK109) and involves ligation of unique barcodes to DNA ends of each sample, followed by pooling and then ligation of sequencing adapters. The LSK and 96 barcoding expansion combination offers higher yield potential, greater control over read lengths, and a lower per-sample costs compared to the RBK kits. Multiplexing 96 bee-collected pollen loads would reduce per-sample costs from the £61 in this study to ~£7.

2.4.4 Application to pollen collected from wild bees

Multiple DNA extraction methods were tested on individual pollen loads collected from wild *Apis* and *Bombus* bees (data not shown), including column-based kits, NucleoSpin Food (Macherey-Nagel, Düren, Germany) and DNeasy Plant (Qiagen, Hilden, Germany). Although the column-based protocols were more convenient to use, we found the traditional CTAB-based phenol chloroform extraction resulted in improved DNA yield and molecule length.

The RevMet pipeline detected consistent differences in the composition of pollen loads collected by honeybees *Apis mellifera* and by the two bumblebees *Bombus terrestris/lucorum* and *B. lapidarius* (Figure 2.4). The low number of plant species identified per pollen load is consistent with the flower constancy behaviour observed in a range of insect pollinators, in which individual pollinators almost exclusively visit a single flower type during a foraging trip (Grüter and Ratnieks 2011). This method can therefore be used to compare flower constancy at individual level between foraging bee species, or in different environmental or seasonal contexts. It is possible to collect bulk pollen samples from managed bee colonies (such as *Apis mellifera* or *Bombus terrestris*) using pollen traps, or pollen samples brought back to trap nests by foraging solitary bees (as for example in Sickel et al. 2015). This would be expected to reveal a much higher diversity of food plants, at least for generalist bee species.

As a proof-of-concept study, only a small number of bee-collected pollen loads (n = 48) sampled from just one site were analysed. By generating more plant reference skims and utilising the 96 native barcode expansion for cheaper multiplexing of pollen loads, RevMet could now be applied to compare pollination networks across large-scale spatial and biogeographical gradients. The RevMet pipeline's ability to assess DNA composition from read counts has been demonstrated. However, there are other potential sources of bias that may have affected our pollen sample proportions, such as the bi- or tri-cellular nature of pollen and differing ploidy levels, genome sizes, and DNA extraction efficiencies. The next step in this research will be to test RevMet's ability to discern different quantities of pollen biomass, and to explore whether it performs better than standard meta-barcoding approaches at this task.

The RevMet pipeline can readily be applied to a wide range of research questions. RevMet could potentially be used to quantify the degree to which co-attraction of pollinators leads not to benefits of increased pollinator numbers but to loss of pollination service via competition (Carvalheiro et al. 2014; Pornon et al. 2016). Outside of pollination ecology, there is potential for semi-quantitative assessments of many other eukaryotic species mixtures, including herbivore diets (Kress et al. 2015; Bhattacharyya et al. 2019); plant-fungus interactions (Schroter et al. 2019); and allergenic pollen species from air samples - although this might require an additional whole-genome amplification (WGA) step (Kraaijeveld et al. 2015). Furthermore, due to the portability and real-time nature of the MinION platform, the method could be optimised for analysis in the field alongside sample collection.

2.5 Data availability

The Illumina and MinION datasets are available in the European Nucleotide Archive (http://www.ebi.ac.uk/ena) under study accession PRJEB30946. Example desktop RevMet scripts are available from https://github.com/nedpeel/RevMet (https://doi.org/10.5281/zenodo.3277268) and a tutorial using an example dataset can be found at https://revmet.readthedocs.io/en/latest/.

3 Chapter 3 – Development of MARTi (Metagenomic Analysis in Real Time)

3.1 Introduction

Metagenomics is transforming our understanding of the diversity and ecology of environmental and clinical microbial communities. Advancements in this field are largely driven by developments in DNA sequencing technology and the associated analysis tools and pipelines. Next-generation sequencing (NGS) technologies, such as the sequencing by synthesis platforms developed by Illumina, have dramatically reduced sequencing costs, which in turn has facilitated metagenomic investigation at a much greater scale and depth. However, low-level taxonomic assignments, highly contiguous metagenomic assemblies, and a rapid time to result are often not achievable using these platforms.

Newer long-read sequencing platforms, such as those developed by Oxford Nanopore Technologies (ONT) and Pacific Biosciences (PacBio) could overcome many of the known problems associated with short-read metagenomics. Significantly, the ONT platform is the first to enable progressive real-time analysis of data. Nevertheless, the full potential of nanopore sequencing remains largely unrealised due to the lack of open source, offline, real-time analysis tools and pipelines. As discussed in Chapter 1, ONT's own EPI2ME platform provides near real-time analysis, but has limitations due to its closed nature. Recognising the need for an open, extensible platform, we developed MARTi (Metagenomic Analysis in Real-Time), an open-source software tool that enables real-time analysis and visualisation of metagenomic sequencing data. MARTi provides a rapid, lightweight web interface that allows users to understand community composition and identify antimicrobial resistance (AMR) genes in real time.

This chapter provides an overview of the MARTi tool and its implementation, and is comprised of the following sections:

3.2 Architecture - an overview of MARTi's overall architecture, which is comprised of two components – the MARTi Engine and the MARTi GUI.

3.3 MARTi Engine - a brief description of the MARTi Engine, the back end developed by Richard Leggett.

3.4 MARTi GUI - a technical description of the MARTi GUI, which was the main focus of this thesis.

3.5 Samples page – a breakdown of the features of the GUI's Samples page that used for selecting and loading samples into MARTi's analysis modes.

3.6 Dashboard mode – a detailed description of MARTi GUI's Dashboard analysis page, used for viewing a single sample.

3.7 Compare mode – a description of the Compare page, which enables users to compare multiple samples.

3.8 New analysis page - a description of the page used to configure and start a local MARTi analysis from the MARTi GUI.

3.9 Tree methods - algorithmic descriptions are provided for some of the key methods used to process the Dashboard mode's taxonomic tree.

3.10 Summary and future work – a summary of the chapter and comments on further development.

In the next chapter, we present results showing the use of MARTi in real sequencing applications.

3.2 Architecture

MARTi consists of two main components: the MARTi Engine (developed by Richard Leggett), which is a command-line initiated Java back-end that performs pre-processing and classification of sequence data; and the MARTi GUI (developed for this thesis), a lightweight browser-based front-end for visualising, exploring and comparing results. This modularity allows MARTi to be installed and operated in several different ways depending on the needs of the experiment and computational equipment available. The two most common configurations of MARTi, Local and HPC, are described in the following sections (Figure 3.1).

3.2.1 Local configuration

In local configuration, both the MARTi Engine and the MARTi GUI are installed on a single laptop/desktop (Figure 3.1a). This configuration does not rely on any external computing resources and can therefore be operated without an internet connection, making it suitable for analysis in-field or in-situ. A nanopore sequencing device, such as a MinION Mk1C or GridION, generates batches of basecalled reads. Data is transferred to the MARTi computer either by mapping the sequencer's drive, or (with greater reliability) via an rsync process to synchronise the data to the local disc. The MARTi Engine analysis process can then be initiated through the command line or via the MARTi GUI. In this configuration, the MARTi GUI server is run on the same computer as MARTi Engine and provides the analysis results to any connected web browser. This could be a browser on the same computer or any other device on the same network, including tablets and mobile devices.

3.2.2 HPC configuration

The HPC (High Performance Computing) configuration allows users to run more analysis processes in parallel by making use of an additional server or compute cluster to take some of the analysis load (Figure 3.1b). This reduces the gap between the analysis and sequencing rates and enables analysis of multiple runs simultaneously or facilitates queries against larger databases. In this configuration, the MARTi Engine runs on an HPC or separate server, whilst the MARTi GUI resides elsewhere, for example on a laptop/desktop. The chunks of basecalled reads produced by the sequencing device are synchronised to a network drive. The MARTi Engine analyses the data on the HPC and generates output files for the front end. The MARTi GUI's server is run on a desktop/laptop that has access to the network location containing the output files. The server processes and serves the data to any connected web browsers, which could be on the same computer, or any other web browsing device on the same network.

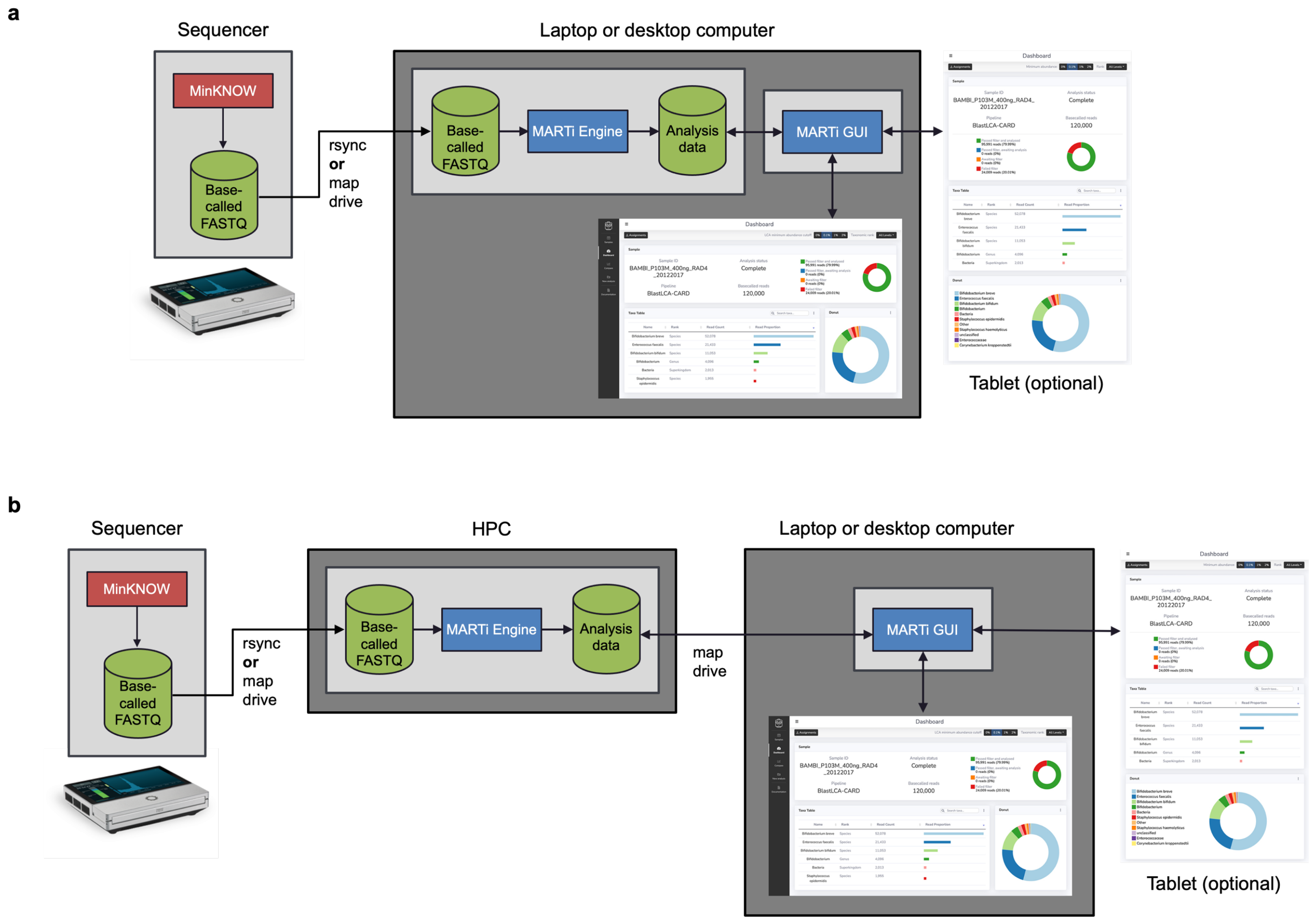


*Figure 3.1 Two main configurations of MARTi. **a** – In local configuration, the MARTi Engine and the MARTi GUI are installed on a single laptop/desktop. **b** – In HPC configuration, the MARTi Engine runs on an HPC or separate server, whilst the MARTi GUI resides elsewhere, for example on a laptop/desktop.*

3.3 MARTi Engine

The MARTi back-end, or MARTi Engine is responsible for processing and analysing the basecalled reads generated by the sequencing device. The MARTi Engine carries out the following processes:

- Prefiltering - Basecalled reads first pass through a prefilter that removes low quality or short reads based on user-set thresholds. The default minimum length is 500 bp in order to remove short reads, which have low taxonomic discriminatory power, and nanopore adapter sequences, which can cause incorrect assignments to poor quality reference genomes that contain these sequences. The default minimum mean quality score is 9, equal to the pass read cutoff used by ONT's Guppy basecaller when using the HAC model. Reads that pass filtering are batched into chunks of a specified size for further analysis. The division of reads into chunks permits parallelisation of the later stages.
- Classification - MARTi classifies reads with a combination of BLAST and its own Lowest Common Ancestor (LCA) algorithm (see below) to assign reads to taxa based on the BLAST results. This algorithm assigns reads to the lowest taxonomic level consistent with "good" hits. The definition of good is configurable but depends on the BLAST bit score, length of match, percent identity of the match and the maximum number of hits to consider.
- AMR analysis – If specified in the configuration file, the MARTi Engine will also BLAST the filtered reads to CARD, the Comprehensive Antibiotic Resistance Database (Alcock et al. 2020), to identify antimicrobial resistance (AMR) genes. The host species of an AMR gene hit can sometimes be identified using the taxonomic classification that was given to the read via the BLAST and LCA pipeline. This process is known as walkout analysis (Leggett et al. 2020) as it often relies on the flanking DNA sequences of the AMR gene for low-level

taxonomic assignments. If a read is not long enough to contain flanking regions, it is more likely have hits to multiple species and therefore be assigned to a higher taxonomic level by the LCA algorithm. Similarly, AMR genes based on plasmids can pose a challenge as the flanking regions can often have ambiguous taxonomic hits.

- Generating output – The Engine writes out analysis data and output files, including those required for the MARTi GUI to function.

3.3.1 Lowest Common Ancestor algorithm

MARTi implements a Lowest Common Ancestor algorithm as follows:

1. Reads are BLASTed against a user defined database. This may be, for example, the whole of NCBI nt, a bacteria subset, RefSeq genomes or a custom database. This results in a set of between 0 and many hits for each read.
2. For a given read, the set of "good hits" is identified by finding the highest scoring hit (according to the BLAST bitscore), then finding all hits with a score within 90% (default value, but configurable) up to a limit (default 20 hits, but configurable).
3. For each good hit, the taxonomic path is determined by referring to the NCBI taxonomy. For example:

   `"root, cellular organisms, Bacteria, Proteobacteria, Gammaproteobacteria, Enterobacterales, Enterobacteriaceae, Klebsiella, Klebsiella pneumoniae"`

The taxonomic paths for all good hits are compared to determine the common ancestor. This involves starting at the root node and working downwards, comparing nodes (first "root", then "cellular organisms", then "bacteria" etc.) until paths diverge.

The last node in common before paths diverge is the common ancestor and the read is assigned to this taxon.

Since MARTi classifies reads with a combination of BLAST and an LCA algorithm, it produces similar classification results and has the same taxonomic resolution as other tools based on this approach, such as MEGAN. When compared to the k-mer based classification tool Kraken2, the classification success (the ratio of correctly classified reads to all classified reads) of a BLAST-based approach for nanopore reads is superior across all kingdoms of life (Pearman et al. 2020). The MARTi Engine writes out MEGAN read-match archive (RMA) files, allowing MARTi's classification results to be explored in MEGAN.

3.4 MARTi GUI

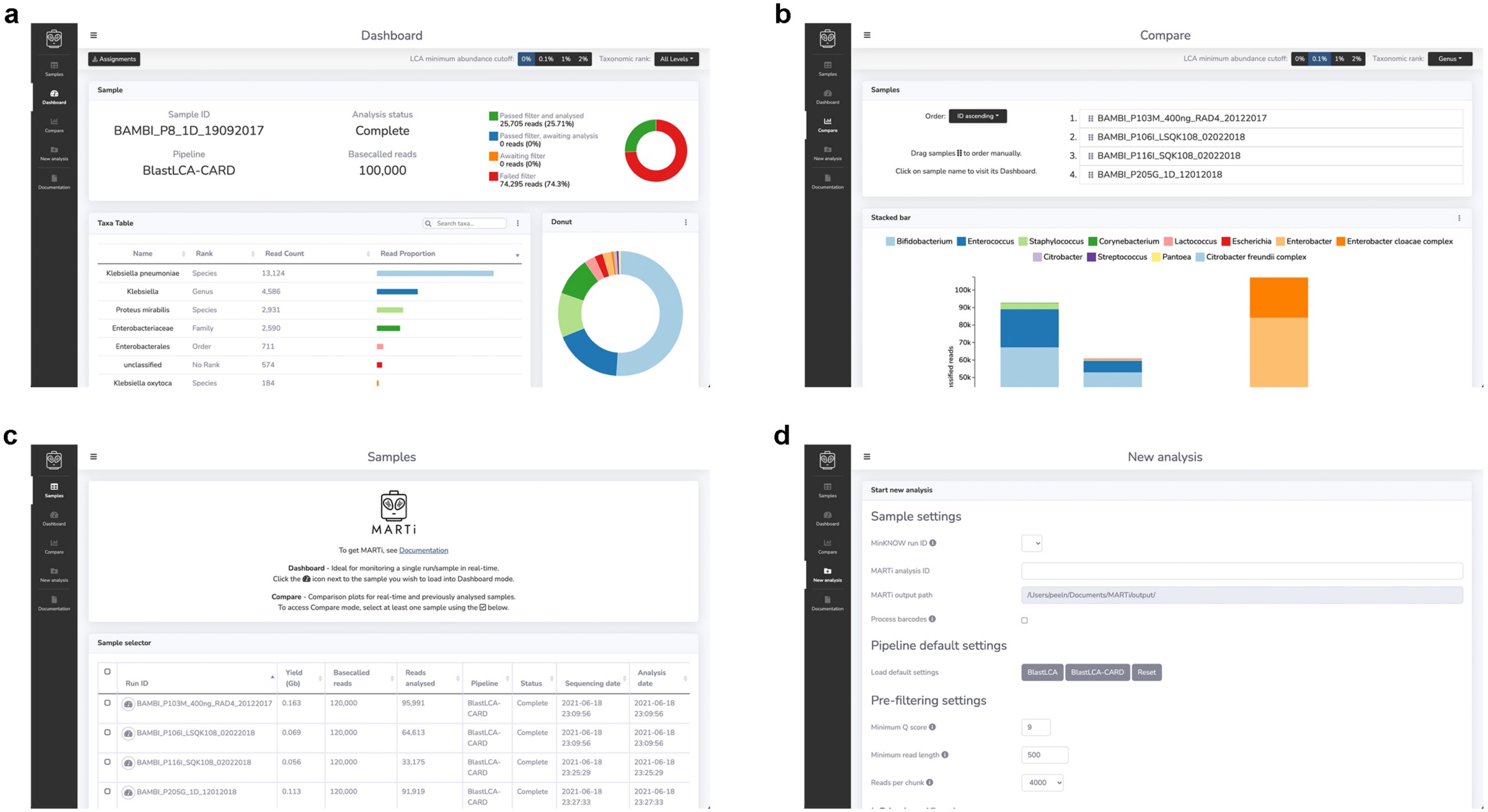
The MARTi GUI is a lightweight browser-based frontend that allows users to view and interact with their results (Figure 3.2). The GUI has four pages:

1. *Dashboard* – enables users to view metrics and real-time analysis results of a single sample (Figure 3.2a).
2. *Compare* - allows users to compare the results from multiple samples (Figure 3.2b).
3. *Samples* - for selecting and loading available samples into the *Dashboard* and *Compare* modes (Figure 3.2c).
4. *New analysis* - allows users to configure and start a local MARTi analysis from the MARTi GUI (Figure 3.2d).

The content and implementation of each of these pages will be discussed in detail in the following sections.

Compared to traditional desktop applications, browser-based applications have several advantages: they can operate without installation across various operating systems and devices via a web browser, they can be accessed by multiple devices on a local network simultaneously, and the UI can be customised for various screen sizes by utilising responsive design.

The GUI is written using Hypertext Markup Language (HTML), Cascading Style Sheets (CSS) and JavaScript (JS). HTML forms the layout and structure of the interface whilst CSS controls how those HTML elements are displayed. JS is used for generating plots ‘on the fly’ and making the UI interactive. MARTi also makes use of Bootstrap (Otto and Thornton 2019), an open-source web front-end framework that makes it easier to develop responsive web content, tailoring the experience for different devices.



Dashboard

Sample

Sample ID
BAMBI_P8_1D_19092017

Pipeline
BlastLCA-CARD

Analysis status
Complete

Basecalled reads
100,000

🟩 Passed filter and analysed
25,705 reads (25.71%)

🟦 Passed filter, awaiting analysis
0 reads (0%)

🟧 Awaiting filter
0 reads (0%)

🟥 Failed filter
74,295 reads (74.3%)

Taxa Table

Name	Rank	Read Count	Read Proportion
Klebsiella pneumoniae	Species	13,124	
Klebsiella	Genus	4,586	
Proteus mirabilis	Species	2,931	
Enterobacteriaceae	Family	2,590	
Enterobacterales	Order	711	
unclassified	No Rank	574	
Klebsiella oxytoca	Species	184	

Compare

Samples

Order: ID ascending

Drag samples ⠿ to order manually.

Click on sample name to visit its Dashboard.

1. BAMBI_P103M_400ng_RAD4_20122017
2. BAMBI_P106I_LSQK108_02022018
3. BAMBI_P116I_SQK108_02022018
4. BAMBI_P205G_1D_12012018

Samples

To get MARTi, see Documentation

Dashboard - Ideal for monitoring a single run/sample in real-time.
Click the icon next to the sample you wish to load into Dashboard mode.

Compare - Comparison plots for real-time and previously analysed samples.
To access Compare mode, select at least one sample using the ☑ below.

Sample selector

□	Run ID	Yield (Gb)	Basecalled reads	Reads analysed	Pipeline	Status	Sequencing date	Analysis date
□	BAMBI_P103M_400ng_RAD4_20122017	0.163	120,000	95,991	BlastLCA-CARD	Complete	2021-06-18 23:09:56	2021-06-18 23:09:56
□	BAMBI_P106I_LSQK108_02022018	0.069	120,000	64,613	BlastLCA-CARD	Complete	2021-06-18 23:09:56	2021-06-18 23:09:56
□	BAMBI_P116I_SQK108_02022018	0.056	120,000	33,175	BlastLCA-CARD	Complete	2021-06-18 23:25:29	2021-06-18 23:25:29
□	BAMBI_P205G_1D_12012018	0.113	120,000	91,919	BlastLCA-CARD	Complete	2021-06-18 23:27:33	2021-06-18 23:27:33

New analysis

Start new analysis

Sample settings

MinKNOW run ID

MARTi analysis ID

MARTi output path: /Users/peeln/Documents/MARTi/output/

Process barcodes □

Pipeline default settings

Load default settings

Pre-filtering settings

Minimum Q score: 9

Minimum read length: 500

Reads per chunk: 4000

Figure 3.2 The four pages of the MARTi GUI. ***a*** *– The Dashboard page enables users to explore real-time analysis results of a single sample.* ***b*** *– The Compare page allows users to compare results from multiple samples.* ***c*** *– The Samples page is for selecting available samples for Dashboard and Compare modes.* ***d*** *– The New analysis page allows users to generate a configuration file and start a local MARTi analysis.*

3.4.1 Node.js

The MARTi GUI is built as a Node.js application that sits between the MARTi Engine and the user’s browser. Node.js, is an open-source, cross-platform, runtime environment built on Google Chrome's V8 JavaScript engine (OpenJS Foundation 2020). JavaScript engines were originally only found in web browsers, with all major browsers having their own built-in JS engines to execute JavaScript code on the client device. More recently, runtime environments such as Node.js have been developed that allow JavaScript to be executed outside of the web browser.

Node.js was developed to facilitate the creation of real-time web applications and unify web application development around a single language. Servers built with Node.js utilise an event-driven, non-blocking I/O model. Therefore, Node.js servers, which run as single-threaded applications, use asynchronous function calls to maintain concurrency. This allows them to remain lightweight and efficient, making them ideal for data-intensive real-time applications. Many fast and I/O intensive real-time applications have been developed with Node.js, including communication apps such as Slack and Skype, as well as MinKNOW, ONT’s own sequencing software.

The node package manager, known as npm, is included in the Node.js installer. The npm tool allows users to access and distribute JavaScript modules via an online repository. MARTi utilises several npm modules:

- Express.js - Express is the most popular web application framework for Node.js (Wilson 2018). It provides mechanisms to handle common web-development tasks such as handling requests to different URL paths ("routes") and serving static files, including images, HTML, CSS, and JavaScript files.
- Socket.IO – enables real-time bidirectional communication between web servers and clients using TCP/IP socket protocols (Rauch 2018).

- chokidar – a lightweight and efficient cross-platform file watching library (Miller 2020).
- fs-extra - adds file system (fs) methods that aren't included in the native fs module and adds promise support to the fs methods (Richardson 2020).
- uuid – for generating Universally Unique IDentifiers (UUIDs) (Kieffer et al. 2020).

A full list of packages and modules used by the MARTi GUI can be seen in Table 3.1.

When the MARTi node server is launched it reads an options file, marti_server_options.txt, that is configured by the user and contains full paths to four directories:

1. MinKNOWRunDirectory – The directory that contains the nanopore sequencing data output by MinKNOW. This is either a path to the data directory on the mapped sequencing device drive or the directory that sequencing data is being synced to. The node server monitors this directory for new runs to analyse.
2. MARTiSampleDirectory – The MARTi Engine output directory. The server monitors the MARTi output files for updates.
3. BlastDatabaseDirectory – Path to directory of BLAST databases.
4. TaxonomyDirectory – Path to directory of NCBI taxonomy databases.

Table 3.1 Packages used by the MARTi GUI

Package	Version	Description	License	URL
Bootstrap	4.3.1	Front-end web development framework	MIT	https://github.com/twbs/bootstrap/releases/tag/v4.3.1
Node.js	12.14.1	Cross-platform back-end JavaScript runtime environment	MIT	https://github.com/nodejs/node/releases/tag/v12.14.1
Express.js	4.16.4	A flexible web application framework for Node.js	MIT	https://github.com/expressjs/express/releases/tag/4.16.4
Socket.IO	2.1.1	Enables real-time bidirectional communication between web servers and clients	MIT	https://github.com/socketio/socket.io/releases/tag/2.1.1
Chokidar	3.4.2	A lightweight and efficient cross-platform file watching library	MIT	https://github.com/paulmillr/chokidar/releases/tag/3.4.2
fs-extra	9.0.1	Adds additional file system methods to Node.js	MIT	https://github.com/jprichardson/node-fs-extra/releases/tag/9.0.1
uuid	8.3.2	For the creation of RFC4122-specification UUIDs	MIT	https://github.com/uuidjs/uuid/releases/tag/v8.3.2
jQuery	3.3.1	JavaScript library for HTML DOM manipulation, event handling, CSS animation, and Ajax	MIT	https://github.com/jquery/jquery/releases/tag/3.3.1
D3.js	3.5.17	JavaScript library for producing dynamic and interactive data visualisations	BSD	https://github.com/d3/d3/releases/tag/v3.5.17
Font Awesome Free	5.8.1	Font and icon toolkit based on CSS and Less	MIT, SIL OFL, CC	https://github.com/FortAwesome/Font-Awesome/releases/tag/5.8.1
DataTables	1.10.19	A jQuery plug-in for advanced HTML table features	MIT	https://github.com/DataTables/DataTables/releases/tag/1.10.19
Google Fonts (Nunito)	-	A sans serif typeface superfamily	SIL OFL	https://fonts.google.com/specimen/Nunito
SortableJS	1.13.0	A JavaScript library for reorderable drag-and-drop lists	MIT	https://github.com/SortableJS/Sortable/releases/tag/1.13.0
canvg	3.0.7	JavaScript SVG parser and renderer on Canvas.	MIT	https://github.com/canvg/canvg/releases/tag/v3.0.7
ResizeSensor.js	0.2.4	JavaScript library for efficient element resize detection.	MIT	https://github.com/procurios/ResizeSensor/releases/tag/0.2.4

3.4.2 Client-server communication

The MARTi GUI consists of a set of resources, including HTML, CSS, and Javascript files, that must be passed to the browser client. In order to access these resources, the client needs to request them from the server. HTTP (Hypertext Transfer Protocol) is used to structure requests and responses between browsers and web servers. The HTTP request and response, including resources, are transferred via TCP (Transmission Control Protocol).

The REST (REpresentational State Transfer) architectural style is the most common way of structuring a web service for requests. Web services adhering to REST principles are described as RESTful. In general, RESTful services use HTTP as the underlying communication protocol and a request-response model where the client requests a resource and then the server responds. For these cases, HTTP methods such as GET, POST, PUT, and DELETE are used to retrieve, submit, update, and delete respectively.

HTTP is a stateless protocol, meaning each request is executed independently, without any knowledge of prior requests, even from the same client. Each time an HTTP request is made, a new TCP connection is opened between the server and client, and then terminated again after a response is received. This means that for every request-response cycle the HTTP header and metadata information is sent again. This, combined with the opening and closing of the TCP connections and processing of the additional information each cycle results in additional latency in cases of frequently repeated request-response cycles such as applications that want rapid responses or real time interactions or display streams of data.

Another limitation of HTTP is that it is unidirectional, so the browser can request information from the server, but the server can’t send data to the browser when it wants to. This means that browsers have to poll the server for new information by repeating requests every so often to see if there is anything new.

A newer alternative to REST is the WebSocket protocol. WebSocket communication begins with an initial HTTP handshake, but then is upgraded to follow the WebSockets protocol. With WebSockets, a persistent connection between the client and server is established. This connection is full duplex, meaning it's capable of two-way simultaneous communication. Therefore, the server is capable of initiating communication and can push data to the client when new data becomes available.

The MARTi GUI uses the Socket.IO npm module for most of the client-server communication, a module that primarily utilises the WebSocket protocol, but also has a polling fallback option. By using WebSockets, the MARTi server can immediately inform clients when updates become available. However, HTTP is sufficient in cases where a resource needs to be fetched infrequently or just once and the client doesn't want or require ongoing updates. Therefore, when a client first connects to the MARTi server, an HTTP GET request is made for the static files. Following this a WebSocket connection is established and the client registers with the server.

3.4.3 Output files for MARTi GUI

The MARTi Engine generates all of the data required by the GUI. For each MARTi-analysed sample, the Engine creates a directory of data files for the GUI comprised of four types of JSON file and one csv:

1. *sample* – The sample JSON file contains information about the analysed sample such as sample ID, read and yield metrics, classification metrics, analysis status, and details of the analysis performed by MARTi.
2. *tree* – Taxonomic classification data as a tree structure in JSON format. This data is used by the GUI to plot taxonomic figures on the Dashboard and Compare pages. The structure of this file is discussed in more detail further on.

3. *accumulation* – Cumulative count of unique taxa discovered per analysed read chunk at each taxonomic level. The accumulation file is used for plotting taxonomic discovery curves at different taxonomic ranks.
4. *amr* – Contains the AMR gene and walkout analysis data required to plot AMR figures. The AMR file format is discussed in detail further on.
5. *assignments* – A CSV file containing a summary of the taxonomic assignments made by MARTi.

In order to reduce computational load on the GUI and improve response time to user input, the files based on MARTi's taxonomic assignments, the accumulation, tree, and assignments files, are pre-computed at four different LCA minimum abundance cut-off values (0, 0.1, 1, and 2%).

3.4.4 Client registration

Clients must be individually recognised by the server in order to receive the correct data in cases where there are multiple clients simultaneously viewing results on the same MARTi server. When the MARTi GUI is served to a client's browser, the browser emits a registration request to the server with the following pieces of information:

- UUID – this will be set to "null" unless the client has previously been connected to the server.
- Current dashboard sample name – ID of currently selected Dashboard mode sample, or empty string if none selected.
- Compare sample name array – A list of samples selected for Compare mode, or empty array if none are selected.

If the UUID received by the server is null, then a new random version 4 UUID will be generated for the client by the uuid module. A new property is then created in a client-side object to store information about selected samples. Following this, the client is subscribed to a unique socket channel to allow direct communication with that client and a response containing the generated UUID is returned via this channel. The client is now registered and will use this ID in all further requests to the server.

3.4.5 AJAX content

The MARTi GUI has a single true HTML page, the index page, that forms the overall structure of the interface. This page houses the ubiquitous features such as the header, sidebar, footer, and content area. A technique known as AJAX, short for Asynchronous JavaScript and XML, is used to dynamically load other HTML files into the content area of the index page without the need to reload the page. MARTi uses the jQuery AJAX methods to achieve this. jQuery is small JS library that simplifies manipulation of the Document Object Model (DOM), the browser-generated tree of HTML objects, and is frequently used for event handling, animation, and AJAX (The jQuery Team 2018).

The GUI has four dynamically loaded pages that are accessed by clicking items in the sidebar. These are the sample selection page, two analysis mode pages, Dashboard and Compare, and the start new analysis page. When an item on the GUI sidebar receives a click, jQuery handles the event and triggers a function responsible for updating the page title text, loading new content via AJAX, and then scrolling to the top of the page.

The content, such as plots, tables, and forms, in each of the dynamically loaded pages are housed in size-changing Bootstrap content containers called cards. The cards are used in conjunction with Bootstrap's grid system so that the proportions of the cards, and content within them, can be customised based on the user’s screen size.

3.4.6 D3.js

Both of MARTi's analysis modes feature sets of interactive real-time visualisations created with the D3.js (D3) library, which makes use of Scalable Vector Graphics (SVG), HTML, and CSS standards to facilitate the production of browser-based dynamic and interactive data visualisations (Bostock 2016). D3 works by manipulating the DOM in real-time. In order to produce visualisations, data is bound to graphical elements of the DOM. When new data points are introduced, D3 creates associated SVG elements. When data points are removed, D3 deletes the connected DOM elements. When data changes, the associated SVG elements are updated accordingly.

3.5 Samples page

The Samples page allows users to select and load available samples into Dashboard and Compare analysis modes. The page consists of two cards: an information card, which introduces the user to the Dashboard and Compare modes and explains how sample selection works; and the sample table card, which features a dynamically updated sample selection table and a "Compare samples" button.

After the Samples page HTML has been loaded into the content area, a Samples page initialisation function, initialiseSamplePage(), is called. This function has three main roles: the first is to create a table using DataTables, a plug-in for the jQuery Javascript library (SpryMedia 2018); the second is to emit a request for sample data to the server, via the WebSocket protocol, to populate the table; and finally, to attach an event handler to the "Compare samples" button.

After receiving a client's request for sample data, the server emits a response specifically to the requesting client with information about the available MARTi samples in JSON format. Upon receiving the response, the client calls a function, updateSampleTable(), to update the Sample page table with the new data. The function removes all existing rows, if any, from the table, before looping through the samples in the data and adding a new row for each. After adding all of the rows to the table internally, the table is redrawn for a visual update. And finally, the function finishes by attaching event handlers to clickable elements of the new rows including the checkboxes and Dashboard icons.

The client then emits two socket requests to the server, one for the current Dashboard mode sample, and another for the array of selected Compare mode samples. If a Dashboard sample hasn't been selected, the sever responds with an empty string and the Dashboard mode page remains locked. On the other hand, if the server returns a sample ID the colour of the Dashboard icon of the matching sample row in the table is changed to green and Dashboard mode is unlocked. Similarly, if no samples have been selected for Compare mode, then the server responds with an empty array and Compare mode remains locked. If the server responds with an array of sample IDs, then the associated sample rows in the table are highlighted, the checkboxes are ticked, and Compare mode is unlocked.

When the user selects a Dashboard sample, either by clicking on the run ID or the neighbouring Dashboard icon, the client sends the ID of the selected sample to the server. When the server receives the ID, it updates the server-side client information object, recording the current Dashboard selection for that particular client. The server responds by sending a *current-dashboard-sample-response* message back to the client, unlocking the Dashboard page if previously locked. The client's content area is then switched to Dashboard mode.

When a user adds or removes a sample for comparison by clicking a rows checkbox, a variable containing the compare sample array is updated and then sent to the server. The server updates the client information object and emits *current-compare-*

samples-response back to the client. If the compare sample array is empty the Compare mode is locked, otherwise it's unlocked.

If a new MARTi analysis sample is added or an existing sample is updated, the server will let all connected clients know that there has been an update. The client has page-specific responses to some server messages including this one. If a client is on the Samples page when it receives this message then it will send a sample data request to the server, as it did when the Samples page was initialised. As before, the rows of the table are internally destroyed, rows are added based on the received data, and then the table is visually redrawn. From the user's perspective, the table changes seamlessly.

3.6 Dashboard mode

The Dashboard page is designed for viewing analysis results of an individual MARTi sample. This could be a single nanopore sequencing run or an individual barcoded sample within a run. The sample can be one that was previously analysed by the MARTi Engine, or one that is currently being analysed. In the latter event, the information on the page will update in real time when new analysis information is made available by the Engine.

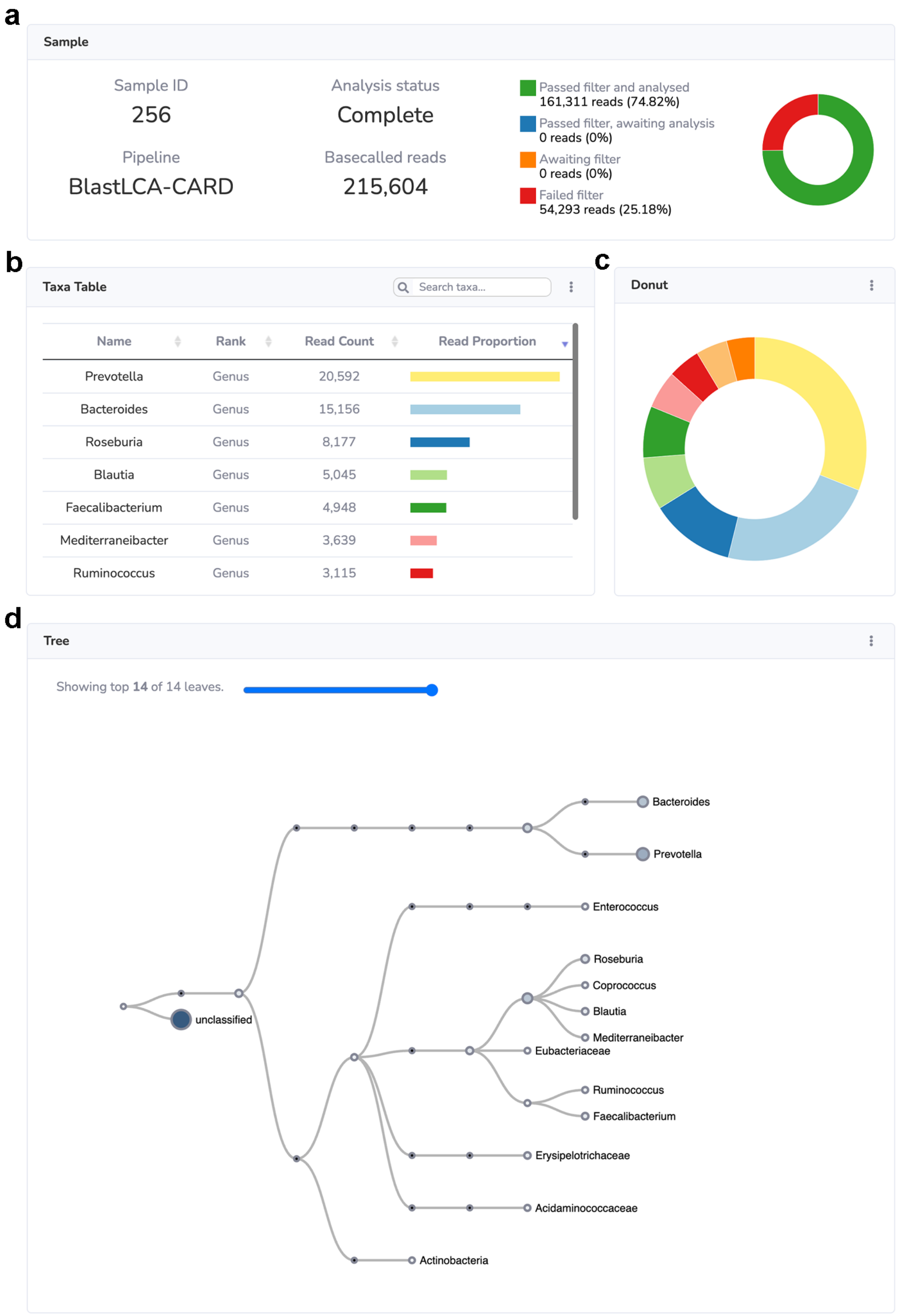
The Dashboard mode content is flexible and dependent on the available analyses for the selected sample. When all available analyses are run for a sample, the page can feature up to 7 cards:

1. Sample card – Displays information about the selected sample such as its ID, the analysis pipeline used, analysis status, and total number of basecalled reads (Figure 3.3a).
2. Taxa table card – A table showing taxa with at least one assigned read at the selected LCA cut off value and taxonomic rank (Figure 3.3b).

3. Donut card – Interactive donut plot of classified reads at selected filter levels (Figure 3.3c).
4. Tree card – Customisable tree plot representing all of the analysed reads, including unclassified (Figure 3.3d).
5. Taxa accumulation card – Features a line chart showing taxa discovered over reads analysed, with an option to switch the x-axis to time (Figure 3.3e).
6. AMR Table card – A table of AMR genes found in the sample (Figure 3.3f).
7. Walkout Analysis card – Donut plot showing results from AMR gene walkout analysis (Figure 3.3g), which involves aligning flanking DNA sequences to identify the host bacteria, as described in (Leggett et al. 2020).

The Dashboard page also features an options bar fixed to the bottom of the header bar. This houses three buttons: the first is for downloading a csv of taxonomic assignments, the second is for selecting the LCA minimum abundance cut-off value, and the last one is a dropdown that allows users to select a taxonomic rank.

After the Dashboard HTML has been fully loaded into the main content area, the dashboard initialisation function, initialiseDashboardPage(), is called. This function carries out four main tasks: it emits requests to the server for sample information, taxonomic classification data, accumulation data, and AMR data; initialises empty tables in the Taxa Table and AMR Table cards using DataTables; attaches event handlers to search boxes and buttons; and finally, calls the initialisation functions for each of the D3 plots.



Sample

Sample ID
256

Analysis status
Complete

Pipeline
BlastLCA-CARD

Basecalled reads
215,604

🟩 Passed filter and analysed
161,311 reads (74.82%)

🟦 Passed filter, awaiting analysis
0 reads (0%)

🟧 Awaiting filter
0 reads (0%)

🟥 Failed filter
54,293 reads (25.18%)

Taxa Table

Search taxa...

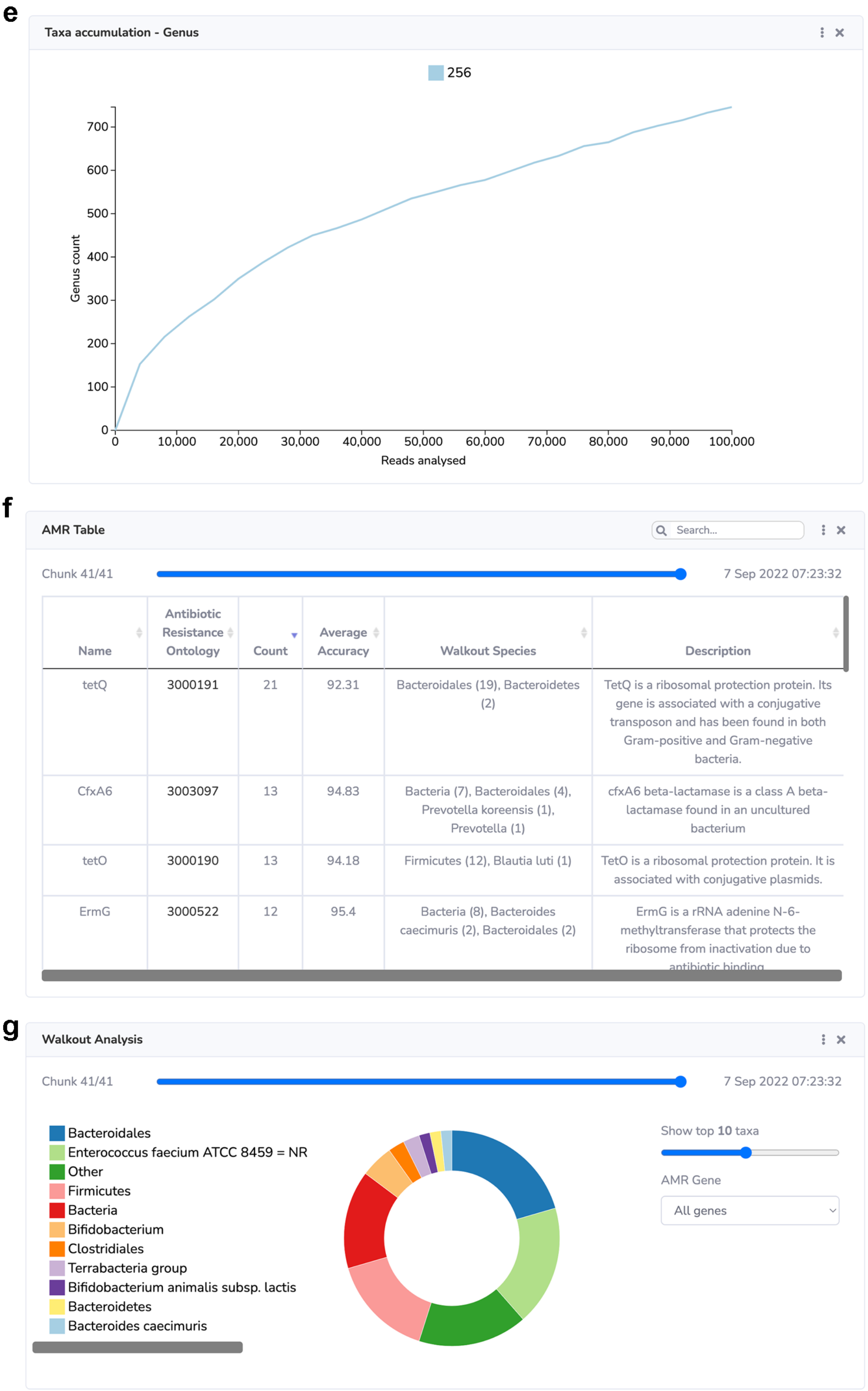
Name	Rank	Read Count	Read Proportion
Prevotella	Genus	20,592	
Bacteroides	Genus	15,156	
Roseburia	Genus	8,177	
Blautia	Genus	5,045	
Faecalibacterium	Genus	4,948	
Mediterraneibacter	Genus	3,639	
Ruminococcus	Genus	3,115	

Donut

Tree

Showing top **14** of 14 leaves.

Figure 3.3 The MARTi GUI Dashboard page cards. ***a*** *– Sample card - displays information about the selected sample.* ***b*** *– Taxa Table - shows taxa with at least one assigned read at the selected LCA cut off value and taxonomic rank.* ***c*** *- Donut card - features an interactive donut plot of classified reads at selected filter levels.* ***d*** *– Tree card – Contains an interactive tree plot representing all of the analysed reads.* ***e*** *- Taxa accumulation card - Features a line chart showing taxa discovered over reads analysed.* ***f*** *- AMR Table card - A table of AMR genes identified in the sample.* ***g*** *- Walkout Analysis card - Donut plot showing results from AMR gene walkout analysis.*



Name	Antibiotic Resistance Ontology	Count	Average Accuracy	Walkout Species	Description
tetQ	3000191	21	92.31	Bacteroidales (19), Bacteroidetes (2)	TetQ is a ribosomal protection protein. Its gene is associated with a conjugative transposon and has been found in both Gram-positive and Gram-negative bacteria.
CfxA6	3003097	13	94.83	Bacteria (7), Bacteroidales (4), Prevotella koreensis (1), Prevotella (1)	cfxA6 beta-lactamase is a class A beta-lactamase found in an uncultured bacterium
tetO	3000190	13	94.18	Firmicutes (12), Blautia luti (1)	TetO is a ribosomal protection protein. It is associated with conjugative plasmids.
ErmG	3000522	12	95.4	Bacteria (8), Bacteroides caecimuris (2), Bacteroidales (2)	ErmG is a rRNA adenine N-6-methyltransferase that protects the ribosome from inactivation due to antibiotic binding

Figure 3.3 continued. The MARTi GUI Dashboard page cards. ***a*** *– Sample card - displays information about the selected sample.* ***b*** *– Taxa Table - shows taxa with at least one assigned read at the selected LCA cut off value and taxonomic rank.* ***c*** *- Donut card - features an interactive donut plot of classified reads at selected filter levels.* ***d*** *– Tree card – Contains an interactive tree plot representing all of the analysed reads.* ***e*** *- Taxa accumulation card - Features a line chart showing taxa discovered over reads analysed.* ***f*** *- AMR Table card - A table of AMR genes identified in the sample.* ***g*** *- Walkout Analysis card - Donut plot showing results from AMR gene walkout analysis.*

3.6.1 Sample card

When the server returns the sample information to the client, jQuery is used to select and update text in the Sample card and then the read breakdown donut plotting function is called with the received data. The donut splits basecalled reads into four categories:

1. Passed filter and analysed - these are reads that have met MARTi's filtering requirements, such as minimum quality score and read length, and have gone through a classification process.
2. Passed filter, awaiting analysis - reads that have met MARTi's filtering requirements and await further analysis.
3. Awaiting filter - reads waiting to be filtered by the MARTi Engine.

Failed filter - reads that have failed to meet the filtering requirements and have therefore been excluded from further analyses.

3.6.2 Classification data

Upon receipt of the taxonomic classification data from the server, functions to plot the taxonomic figures - the donut, tree, and table - are called. The classification data is generated by the MARTi Engine as a tree structure in JSON format. Each node of the tree has 7 properties that can be used by the GUI:

1. name - the NCBI taxon name e.g., "name": "Gammaproteobacteria"
2. ncbiID - the NCBI taxonomy ID e.g., "ncbiID": 41294
3. ncbiRank - NCBI taxonomic rank e.g., "ncbiRank": “class”
4. rank – a simplified rank system for filtering on the GUI. The 45 different NCBI ranks are reduced to the 8 major ranks plus sub-species and no rank (Table 3.2). This value is an integer from 0 to 9 e.g., "rank": 4
5. value – number of reads assigned to the node e.g., "value": 37

6. summedValue – the sum of the number of reads assigned to the node and number of reads assigned to all descendent nodes e.g., “summedValue": 589
7. children – an array of child nodes

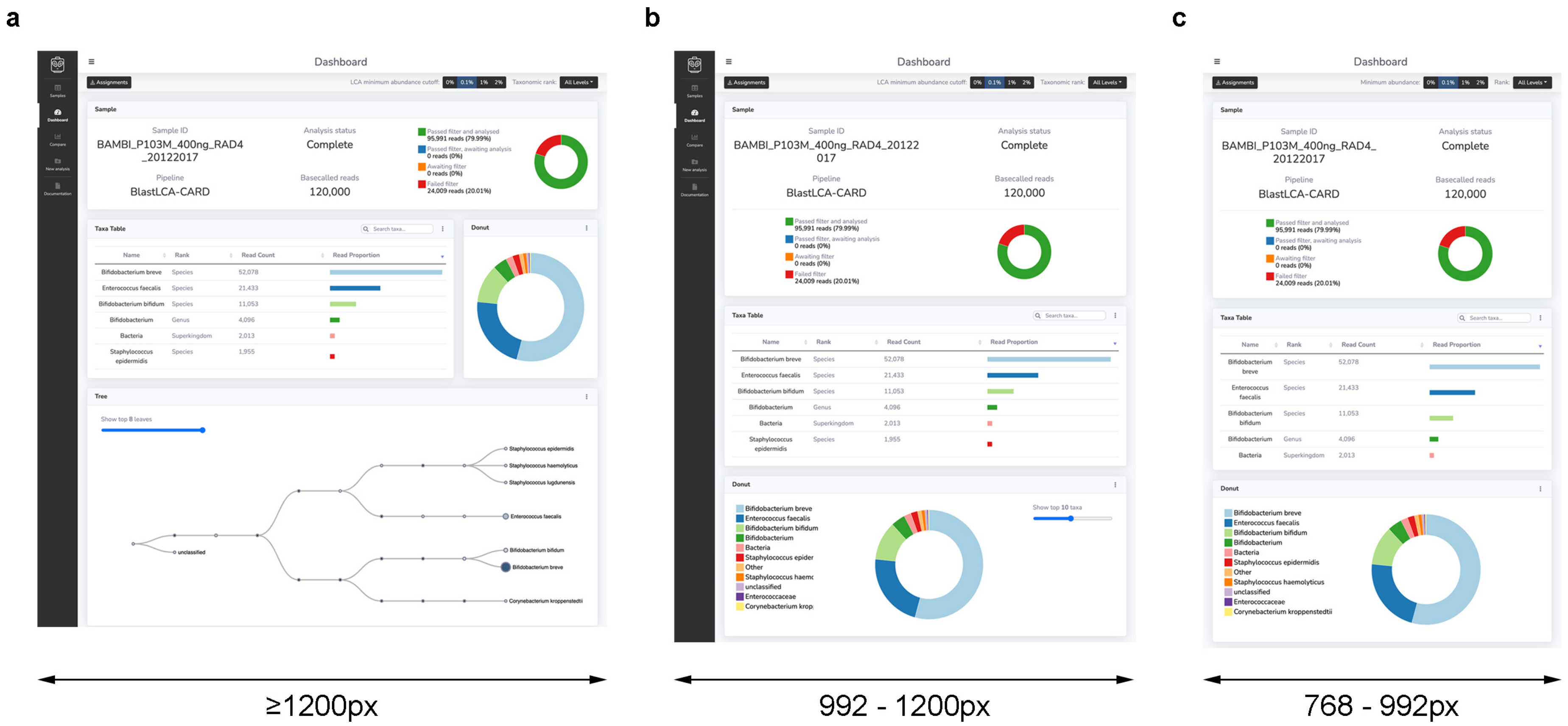
A brief description of how each of these figures are generated from this data will be given here, but some of the key tree functions are explained in more detail in Section 3.9.

Table 3.2 MARTi’s simplified taxonomic rank system

MARTi rank	Rank no.	NCBI ranks
No rank	0	clade, no rank
Domain	1	superkingdom
kingdom	2	kingdom, subkingdom, superphylum
Phylum	3	phylum, subphylum, superclass
Class	4	class, cohort, infraclass, subclass, subcohort, superorder
Order	5	order, infraorder, parvorder, suborder, superfamily
Family	6	family, subfamily, subtribe, tribe
Genus	7	genus, section, series, species group, species subgroup, subgenus, subsection
Species	8	species, genotype, isolate
Subspecies	9	subspecies, biotype, forma, forma specialis, morph, pathogroup, serogroup, serotype, strain, subvariety, varietas

3.6.3 Taxa table and donut plot cards

The taxa table and donut plot share a data processing function and on devices that Bootstrap refers to as extra large (where the width >= 1200 px), the cards sit next to each other on a row underneath the Sample card. In this layout, the table doubles up as an interactive legend for the donut. However, on smaller screens, such as tablets, there isn’t enough space to have the two cards side-by-side. In these cases, the Donut card breaks out onto a new row and two new columns are added to the card, one for the legend and another for the top *n* taxa range slider (Figure 3.4).



Dashboard

Assignments

LCA minimum abundance cutoff: 0% 0.1% 1% 2% Taxonomic rank: All Levels

Sample

Sample ID
BAMBI_P103M_400ng_RAD4_20122017

Analysis status
Complete

Pipeline
BlastLCA-CARD

Basecalled reads
120,000

🟩 Passed filter and analysed
95,991 reads (79.99%)
🟦 Passed filter, awaiting analysis
0 reads (0%)
🟧 Awaiting filter
0 reads (0%)
🟥 Failed filter
24,009 reads (20.01%)

Taxa Table

Name	Rank	Read Count	Read Proportion
Bifidobacterium breve	Species	52,078	
Enterococcus faecalis	Species	21,433	
Bifidobacterium bifidum	Species	11,053	
Bifidobacterium	Genus	4,096	
Bacteria	Superkingdom	2,013	
Staphylococcus epidermidis	Species	1,955	

Donut

Tree

Figure 3.4 The Dashboard page of the MARTi GUI at different viewport widths. The GUI uses Bootstrap's grid system as part of its responsive design. ***a*** *– On viewports wider than or equal to 1200px the Taxa table card and Donut card sit on the same row and table doubles up as an interactive legend for the donut.* ***b*** *– When the viewport is between 992 and 1200px, the Donut card breaks out onto a new row and two new columns are added to the Donut card, one for the legend and another for the top n taxa range slider.* ***c*** *– On smaller viewports, 768 to 992px, the third column of the Donut card is hidden. The top n taxa range slider can be accessed via the options menu. The sidebar can also be hidden on smaller screens.*

The JSON data is first converted to a D3 tree object, making it easier to work with. If a specific taxonomic rank has been selected with the taxonomic rank dropdown, then only nodes at that particular rank are kept, otherwise all nodes proceed to the next step. The labelNewLeaves() function is used to make an array of the leaf nodes that can be referred to when setting the value to be plotted for each node. For non-leaf nodes, the value, that is the number of reads assigned directly to that node, is used as the plotting value for the table and donut. However, for leaf nodes, the summed value is used so that the reads assigned to all of the hidden descendent nodes are rolled back up into the nodes at the selected taxonomic level and included in the visualisation. The table has four columns:

1. Name - the NCBI taxon name, which when clicked opens a new tab at the specific taxon page of the NCBI taxonomy browser.
2. Rank - NCBI taxonomic rank.
3. Read Count – summed read count for leaf nodes and node read count for all other nodes.
4. Read Proportion – an SVG rectangle that allows the user to quickly assess each taxa’s proportion of the total reads at the selected rank.

The first step in the Dashboard taxonomic table update is to calculate the proportion of the largest node plotting value to the sum of plotting values. This proportion will be used to calculate the width of SVG bars in the read proportion column of the table. Next, the table is cleared of any existing rows, a new row is created for each node, and the table is drawn. A function is then called to create, and colour, the SVG proportion rectangles, and add event handlers to the rows for the tooltip and row style changes on hover.

No limit has been placed on the number of rows that the taxa table can display as it is scrollable. The donut plot, on the other hand, is less able to display large numbers of taxa as the slices become very thin and difficult for the user to see and interact with. Therefore, the data goes through an extra step of filtering so that only the most abundant taxa are displayed by the donut. The returnTopTaxa() function sorts the nodes by their plotting value and then adds a property called threshold to each node. Threshold is set to the NCBI ID of the taxa if it is to be shown as its own slice or "Other" if not. The default number of taxa slices to be shown is ten, with the other taxa being rolled up into an "Other" category, but this number can be adjusted by the user with a range slider in the options menu on the Donut card. The value for the "Other" slice is calculated as the sum of the plotting values of the constituent taxa.

The taxa donut and associated legend, which is only visible on smaller screen sizes, are plotted using this simplified data and D3 methods. The donut plotting function also adds event handlers to the slices and legend items to allow reciprocal highlighting and tooltip functionality. The donut also has reciprocal highlighting with the taxa table and the colours of the table read proportion bars are updated when the donut slices change colours so that taxon is colour matched between the two figures.

Besides the highlighting and tooltips, the table and donut feature other interactive elements. For the table, each column can be sorted alphanumerically by clicking on the header cells. Rows can be filtered using a search box, with the table being redrawn on every keyup event. The table also features an export to csv option as well as copy to clipboard. The donut features a top *n* taxa range slider and three export options, SVG, PNG and JPG.

3.6.4 Taxonomic tree card

The next card down on the Dashboard page is the Tree card, which houses an interactive taxonomic tree. The plot allows users to easily visualise the distribution of assigned reads across the taxonomic tree and provides options for customisation. Radio buttons in the card's options dropdown allow the user to select between different algorithms for tree type, link type, node size, and node colour. The tree also features click-collapsible nodes, responsive internodal lengths, a filtering range slider to show only the top *n* taxa, and export options.

As with the donut plot, the tree plotting function is triggered by the receipt of the classification tree JSON data from the server or when the user selects a different taxonomic rank or changes the number of top taxa to show. However, unlike the donut plot, the pre-processing of the data differs depending on how the tree plot update function is triggered.

If new data is received from the server or the user has selected a different taxonomic rank for existing data, then the data is first processed by three important functions: resetTreeBranches(), taxonomicRankFilt(), and topNLeaves().

The resetTreeBranches() function resets the tree branches and prepares it for filtering and pruning by uncollapsing any click-collapsed nodes, restoring any branches that had been pruned, and setting the "keep" property of all nodes to false.

The taxonomicRankFilt() function collapses the tree to the selected taxonomic rank. For nodes at a higher taxonomic rank than selected, any ranks hidden by previous taxonomic rank filtering are unhidden. Nodes at the selected rank have their children hidden in a taxonomic filter specific property. In rare cases a tree path might not contain the selected taxonomic rank but will contain nodes at lower levels than selected, and in these cases the nodes are pruned at their parents using the hideSpecificBranch() function. The taxonomicRankFilt() function also keeps count of leaf nodes and calls a function to update the max value of the top leaves range slider.

The third function, topNLeaves(), is responsible for trimming the tree so that it only features the number of leaves specified by the user. A description of how this function works can be seen in the *Tree methods* section.

When the top *n* leaves value is changed by the user, only resetTreeBranches() and topNLeaves() need to be called as the taxonomic rank remains the same. If new tree data is received from the server, and it's not the first time the tree has been drawn, then the click-collapse status of the nodes must be copied from the old tree and applied to the new one with the copyCollapseState() function.

Subsequently, the pre-processed data is converted to either a D3 tree object or D3 dendrogram object depending on which option is selected. The D3's tree layout implements the Reingold-Tilford "tidy" algorithm for constructing hierarchical node-link diagrams, whereas the dendrogram cluster layout produces a node-link diagram that places leaf nodes of the tree at the same depth. The tidy trees are generally more compact than the dendrogram equivalents and can make it easier to see the taxonomic ranks of assigned reads at a glance. However, the cluster layout can improve the view of the leaf nodes as it eliminates the possibility of overlapping labels.

The tree is then plotted using D3 methods to handle how new data points are added, existing ones updated, and how SVGs no longer needed are removed. Event handlers are then added to the nodes to create a tooltip on mouse over. The tree update function ends by recording the collapse status of each node in a JS object that can be used by the copyCollapseState() function when new data arrives to re-collapse any user collapsed nodes.

3.6.5 Taxa accumulation card

The taxa accumulation card sits underneath the tree card and features a line chart showing the number of taxa discovered per read chunk analysed, or over time if selected by the user. These plots typically resemble a logarithmic growth curve, with lots of new taxa being discovered early on in a sequencing run, but then the discovery rate slows and levels off. The plot gives the user an idea of how much diversity there might be left in the sample to capture and how much more sequencing effort it would take to discover it. With nanopore sequencing, it is therefore possible to conclude that a sample's diversity has been captured and to stop sequencing, preserving the flow cell's life for other samples.

The MARTi Engine outputs the data required for the accumulation card in JSON format, one file per LCA setting. Each file has a property for each taxonomic rank, as well as one for all levels combined. Nested within the taxonomic rank properties are two arrays of coordinates, one for number of taxa per read chunk and the other for taxa per time. As with the taxonomic assignment data (the tree JSON files), the server monitors for the addition, update, and removal of accumulation JSON files. Rather than receiving the whole file, which could be quite large, each client request for accumulation data is at a specific taxonomic rank and LCA cutoff.

The dashboard page emits requests for accumulation data in five circumstances: during initialisation of the dashboard page, when the accumulation plot is unhidden if previously closed, on taxonomic rank change, LCA abundance cutoff change, and in response to an accumulation update available message from the server. When the server responds with the requested data, the client calls the accumulation plot update function, or if this is the first time accumulation data has been received for this particular instance of the Dashboard page, the accumulation plot initialisation function is called followed by the update function.

The accumulation chart has two plotting options, x-axis and curve, as well as export options for SVG, PNG and JPG. The x-axis option allows the user to select whether they want to view the discovered taxa per read chunk sequenced or over time analysed. The curve radio toggle input is used to pick the line plotting algorithm, either linear, which connects the points with straight line segments, or monotone, which attempt to create a smoother curve with monotone cubic interpolation.

The accumulation initialisation function creates the main SVG with a viewBox attribute, to define its position and dimensions, and then draws the axes. The accumulation plot update function is responsible for the rest of the content and uses D3 methods. First, the user's x-axis and curve selections are retrieved, and relevant variables are set. The x and y-axis labels are updated, with the former being set to the selected x-axis option, and the latter set to the selected taxonomic rank. The data is used to generate an object that includes the name of the sample and the values, which are the coordinates for the points to be plotted based on the x-axis selection. The legend, lines, and axis ticks are created based on the input data. The final step of the update function is to add mouse event handlers to the plot. On mouseover events the closest point on the line is highlighted with a circle, the taxa value at that point is added next to the point, and a vertical line is plotted so the user can see where the point intersects with the x-axis.

3.6.6 AMR cards

The next two cards on the dashboard page, the AMR table card and AMR walkout analysis card, only appear if AMR data is available for the sample. If specified in the configuration file, the MARTi Engine will BLAST reads to CARD (Alcock et al. 2020) to identify antimicrobial resistance genes. The AMR JSON file produced by the MARTi Engine contains all of the information required to plot both of the AMR cards. The file contains timestamps for each read chunk analysed, the total number count of AMR hits, and most importantly an array of AMR genes that have been detected in the sample. Each of the genes has six properties associated with them:

1. cardId – the AMR genes CARD accession number.
2. name – name of the gene.
3. description – a short description of the gene obtained from CARD.
4. count – key-value pairs of the cumulative number of reads that have been matched to the gene at each chunk. Only includes a key-value pair when the number of hits changes rather adding one every chunk.
5. averageAccuracy – average accuracy of the AMR hits for each chunk that the hit count changed.
6. species - an object containing species identified during walkout analysis. Each species property houses key-value pairs representing the number of assigned reads to the species at each chunk where a change in number has occurred.

The client requests the AMR JSON file from the server in three circumstances: on initialisation of the dashboard page, when either of the AMR cards is unhidden, and in response to an update available message from the server. After receiving the data, the client calls the AMR table update function and the AMR walkout analysis donut plotting function.

The first time the AMR update functions are called, an initiation function is also called, adding event handlers to the chunk selection range sliders. These sliders are at the top of each of the AMR cards and allow the user to view the cumulative AMR results as they were at a particular read chunk, which is useful for determining the timepoint when a particular resistance was detected. As the user drags the range slider, input events are triggered and text on the card is updated so that the user can see what chunk is about to be selected. When the slider is released a change event is triggered, resulting in the table and AMR donut update functions being called.

To reduce the size of the AMR data, the count, average accuracy and species properties do not have values for every chunk. Instead, new data points are only added if the cumulative count has changed. Therefore, the AMR table update function has to identify the values to plot at a particular chunk selection. If there are values at the selected chunk then those will be used, otherwise the highest chunk that is less than the selected chunk will be used.

The table is plotted in the same way as the taxa and sample selection tables. First the table is cleared of all rows, new rows are added, and finally the table is drawn. The table has five columns:

1. name - the name of resistance gene with an HREF attribute featuring the URL of the gene on the CARD website.
2. count – total number of reads that have hits to that gene at the selected chunk
3. average accuracy – chunk specific average accuracy of the AMR hits.
4. walkout species – a coma separated list of the identified host species with the number of hits in brackets.
5. description - a short description of the gene obtained from CARD.

As well as the chunk range slider, the AMR walkout analysis card features a top taxa slider for specifying the maximum number of taxa to show in the donut and a dropdown for AMR gene selection. The AMR walkout analysis plotting function populates the gene selection dropdown with the names of resistance genes that were identified at or before the selected chunk, and then calls the topTaxaAmr() function on the species hits of either a selected gene or all genes.

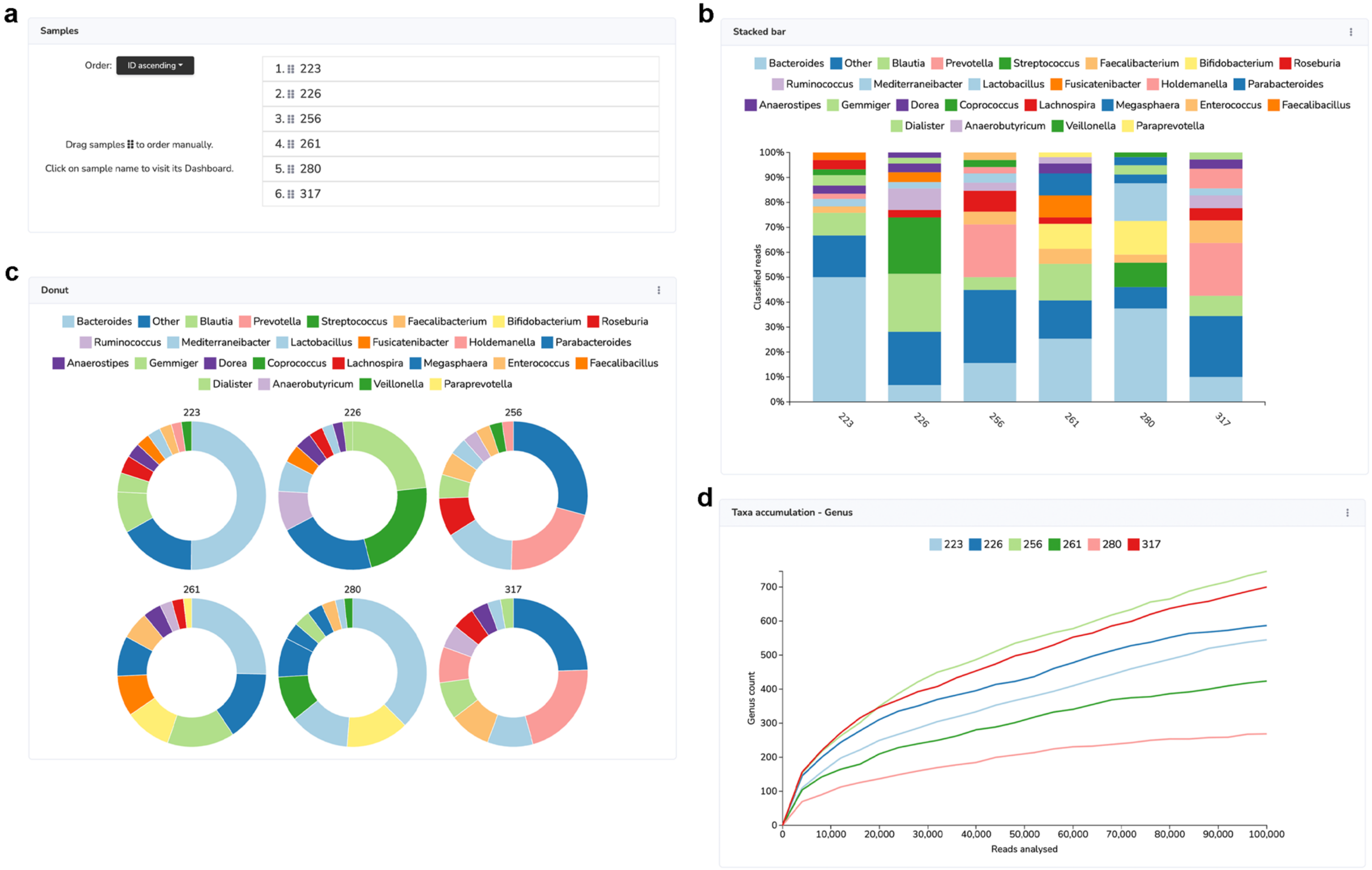
The topTaxaAmr() function takes an array of resistance gene objects and outputs data in a format ready to be plotted as a D3 donut. The code iterates through each identified host taxon for each gene and builds a new array of objects, each containing the name of the taxon and the sum of AMR gene hits. The top taxa range slider value, which by default is 10, is then used to filter the array, keeping the taxa with the top *n* AMR hits and rolling up all of the other taxa into an “Other” category with a summed count. The output data is then used to build an interactive legend and donut plot using D3 methods.

3.7 Compare mode

The Compare mode page allows users to explore multiple samples at once, including samples currently being analysed. This page features four cards:

1. Samples card – Allows the user to sort the selected comparison samples by ID, sequencing date, yield, reads analysed, and by manually dragging them in place (Figure 3.5a).
2. Stacked bar card – A stacked bar chart for viewing the taxonomic composition of the selected samples side-by-side (Figure 3.5b).
3. Multi-donut card – A multi-donut plot for comparing the composition of assigned reads between samples (Figure 3.5c).
4. Taxa accumulation card – A multi-line chart representing taxa discovery rates of each sample over the course of analysis, with the x-axis showing either reads sampled or time analysed (Figure 3.5d).

As with the Dashboard page, the Compare page also has an options bar fixed to the bottom of the header bar. However, the Compare version of this bar only houses two buttons, an LCA minimum abundance cutoff selector, and a taxonomic rank dropdown.



Samples

Order: ID ascending

Drag samples ⠿ to order manually.

Click on sample name to visit its Dashboard.

1. ⠿ 223
2. ⠿ 226
3. ⠿ 256
4. ⠿ 261
5. ⠿ 280
6. ⠿ 317

Figure 3.5 The MARTi GUI Compare page cards. ***a*** *– Samples card – allows users to order the selected samples.* ***b*** *– Stacked bar card - A stacked bar chart for viewing the taxonomic composition of the selected samples side-by-side.* ***c*** *- Donut card - A multi-donut plot for comparing the composition of assigned reads between samples.* ***d*** *– Taxa accumulation – A multi-line chart representing taxa discovery rates of each sample over the course of analysis.*

After the Compare page has been loaded into the main content area, the compare page initialisation function, initialiseComparePage(), is called. This function carries out several important tasks: it adds event handlers to the options bar buttons, Samples card dropdown, and export buttons; calls initialisation functions for the stacked bar and donut compare plots; and emits requests to the server for taxonomic classification and accumulation data.

On receiving a request for taxonomic classification data, the server joins the tree JSON data for all of the selected Compare mode samples together and responds with a single object. When the client receives this taxonomic data, it calls the updateComparePlots() function, a function that prepares the input data and then calls the plotting functions for the two taxonomic compare plots.

The updateComparePlots() function carries out a number of tasks for each of the selected samples: first, the taxa are filtered based on the selection made on the taxonomic rank dropdown in the Compare page options bar; then, the labelNewLeaves() function identifies leaf nodes and a plotting value is set for each node, either as the summed read count for leaf nodes, or read count for all other nodes; the topTaxaCompare() function creates two objects, one for each taxonomic plot, of filtered taxa based on the top *n* taxa value selected for each plot, and also generates an array of taxa names to be included in the plot's legend.

The updateComparePlots() function proceeds by creating an array of the selected sample names and sorting it with a relevant algorithm depending on which option is selected in the Samples card order dropdown. The array of sorted sample names is then used to reorder the draggable sample names in the Samples card to match the dropdown options selected. Finally, the plotStackedBar() and plotCompareDonut() are called with the filtered data and legend arrays.

3.7.1 Stacked bar card

The stacked bar card sits underneath the Samples card and features a stacked bar plot to facilitate taxonomic composition comparisons between samples. The card's options menu features a top *n* taxa per sample range slider, a y-axis radio button for switching between read count and percent, and chart export buttons. When the plotting and update function, plotStackedBar(), is called, D3 methods, including d3.stack(), are used to calculate the position of each taxon on the y-axis for each sample and then create sets of stacked rectangles.

3.7.2 Multi-donut card

The next card present on the Compare page is the Donut card that features an interactive donut for each sample selected for comparison. As with the stacked bar plot, this plot allows the user to compare taxonomic composition between samples. The options menu of the Donut card features four inputs:

1. A top *n* taxa per sample range slider - allows the user to specify the maximum number of different taxa to show in each donut.
2. Donut radius slider - for specifying the max radius of the donuts.
3. Donut area radio buttons - for switching between equal size donuts to area based on the number of assigned reads.
4. Read count radio buttons - for toggling on and off a read count label in the centre of the donut.

The plotCompareDonut() function, which is triggered at the end of the updateComparePlots() function, is used to plot and update the donuts and legend. The function starts by formatting the input data specifically for donut plotting. The user inputs for donut radius and donut area are retrieved and the value of the former is used in the calculation of the latter. D3 methods are then used to plot the legend and donuts, as well as add mouse related event listeners to the legend items and donut slices for reciprocal highlighting. To prevent the donut labels from overlapping, a text wrapping function was developed and is called when the donut name SVG text labels are created.

3.7.3 Taxonomic plot updates

When changes are detected in the tree JSON files for any sample that the MARTi server is watching, the server emits a tree-update-available message to all clients that have the updated sample selected for Dashboard or Compare mode. If the client is on the Compare page, then the client emits a request, compare-tree-request, to the server for the latest Compare mode combined tree data.

3.7.4 Accumulation card

The final HTML card on the Compare page is the Taxa accumulation card that hosts a multi-line chart showing the cumulative number of taxa discovered per read chunk, or over time. This plot is generated using the same function, plotRarefactionCompare(), as the accumulation plot on the Dashboard page and features the same options.

Requests for the Compare page taxa accumulation data are sent to the server in four cases: during the initialisation of the Compare page by the initialiseComparePage() function, when the minimum LCA abundance cutoff is changed, if a different taxonomic rank is selected, and in response to an update available message from the sever. The data returned by server is a single object containing the accumulation JSON data at the specific LCA cutoff and taxonomic rank selected for each of the samples selected for comparison.

3.8 New analysis page

The MARTi Engine requires a configuration file in order to start a new analysis of a run or barcoded sample. The config file provides all the details for the analysis to be performed by the MARTi Engine. A config file can be generated manually by editing a template config file for running on a cluster/HPC environment, or by using the MARTi GUI's new analysis form for running MARTi locally.

The new analysis page allows users to generate a config file and start a local MARTi analysis from the MARTi GUI. The page is comprised of just one HTML card, the Start new analysis card, that houses all of the input fields and buttons required to produce and submit the data required for the server to generate the config file.

When the New analysis page is loaded into the content area, the page's initialisation function, initialiseNewPage(), is called. This function is responsible for four main tasks:

1. Requesting data from the server – During initialisation, the New analysis page requests the server options and the IDs of the available MinKNOW runs from the server by emitting the default-server-options-request message.
2. Generating HTML inputs – As the *Max jobs* and *Select barcodes* inputs have many options, 16 and 96 respectively, it is easier to generate these options with JS rather than creating bloated hard-coded HTML.
3. Attaching event handlers – the "more information" icons and many of the inputs, including some of the text boxes, checkboxes, dropdown menus, range sliders and buttons, need to react to the user's input immediately. For example, when the *Process barcodes* checkbox is ticked, the *Select barcodes* input needs to be revealed. Therefore, the initialisation function adds relevant event handlers to these elements.

4. Handle form validation and submission – the final task of the initialisation function is to define how form validation occurs, which prevents the form from being submitted if the form is incomplete or inputs have values outside of their ranges, and control form submission, which will be discussed in more detail later.

Upon receiving the default-server-options-request from the client, the server scans the MinKNOW run directory (via an additional function written by Samuel Martin) specified in the marti_server_options.txt file for runs to analyse and responds with a serverOptions object. This object contains the four directories from the server options text file, MinKNOWRunDirectory, MARTiSampleDirectory, BlastDatabaseDirectory, and TaxonomyDirectory, as well as an array of any available MinKNOW runs for analysis.

When the client receives the response from the server, the *default-server-options-response* message, it carries out several updates to the page. First, the MinKNOW run ID dropdown is populated with sample names from the minKNOWSampleNames array. Then, the MARTi analysis ID text input field is filled with the name of the first run ID in the dropdown. The un-editable MARTi output path field is updated using the base path from the MARTiSampleDirectory property of the data in addition to the contents of the MARTi analysis ID field. Finally, the client updates the barcode name text fields of all of the unselected barcodes to the MARTi analysis ID and the barcode ID joined with an underscore.

As analysis requirements vary for different users, the MARTi analysis pipeline is customisable. If a user wants to BLAST their sample to the entire NCBI Nucleotide database, then they can click on the *BLAST vs nt* button in the Analysis processes section of the form. In response, a BLAST process card will be added and prefilled with recommended settings for BLASTing to nt. If a user wants to classify their reads using a different database, they can click *BLAST vs other* and specify the database and BLAST settings. Multiple BLAST processes can be added but only one can be used for the LCA classification, signified to the Engine by ticking the *Use this BLAST to classify* checkbox. There is also a *BLAST vs CARD* button that creates a preconfigured analysis process for aligning reads to CARD in order to identify AMR genes.

The form also features buttons to load default settings for the most common MARTi pipelines, BlastLCA and BlastLCA-CARD. The standard BlastLCA pipeline loads default settings for BLASTing to nt followed by MARTi's LCA classification. The BlastLCA-CARD pipeline is for users that also want to identify AMR genes in the sample by an additional BLAST to CARD followed by MARTi's walkout analysis to identify the host organisms of the resistance genes.

A *Reset* button is included in the *Pipeline default settings* section of the form, allowing users to quickly revert all inputs back to default without the need to reload the form page. Having the button next to the pipeline loading buttons makes it easy to revert after browsing different pipeline settings and it is also less likely that a user will accidentally reset the form instead of submitting as can be the case if the reset button is included at the end of the form next to the submit button. When clicked, the reset button calls the resetForm() function that uses jQuery statements to set input element values back to their default values, trigger change and input events if necessary, and remove all analysis process cards.

When the form submission button, *Start analysis*, is clicked, the client checks the validity of form, making sure each element satisfies their constraints. If the form fails validation, then further propagation of the submission event is prevented, an alert pops up to tell the user that form submission has failed, and CSS is used to highlight which form elements passed or failed validation. If the form is valid then the jQuery ajax function is used to submit the form data to the server via an HTTP POST request without reloading the page. When form submission is successful, the user is notified via an alert and then redirected to the samples page.

The data is submitted to the server as a URL-encoded text string where it is converted to JSON by a built-in middleware function of Express that parses all incoming requests with URL-encoded bodies. Server-side code written by Samuel Martin creates a new directory in the MARTi output location with the MARTi analysis ID as its name, writes out a config file using the values from the form data, and then spawns a local MARTi Engine process.

3.9 Tree methods

In this section, algorithmic descriptions are provided for some of the key methods used to process the taxonomic tree that were mentioned in earlier sections.

3.9.1 Collapse tree to selected taxonomic rank

The taxonomicRankFilt() function collapses the tree to the selected taxonomic rank as follows:

1. Set leaf count to 0.
2. Recurse down tree from root towards leaves. For each node:
    1. If taxonomic rank is less than the selected rank:
        1. Unhide any hidden child nodes.
        2. Add 1 to leaf count if node has no children.
    2. If taxonomic rank is equal to the selected rank:
        1. Hide any child nodes.
        2. Add 1 to leaf count.
    3. If rank is greater than the selected taxonomic rank:
        1. Add node to a hide branch list.
3. For each node in the hide branch list:
    1. Hide node and all of its descendants from parent node with the hideSpecificBranch() function.
4. Set maximum value of the top *n* leaves range slider to leaf count.

3.9.2 Prune specific branch

The hideSpecificBranch() function prunes a specific branch from a node when given the name of the child to prune:

1. Create undefined *splice index* variable.
2. For each of the node's children:
    1. If child's name matches name of node to prune:
        1. Create property to store pruned branches unless it already exists.
        2. Add child node to pruned branch property.
        3. Set *splice index* to index of child.
3. Remove child from *splice index* position of children array.

3.9.3 Trim tree to show top n leaves

The topNLeaves() function is responsible for trimming the taxonomic tree so that it only features the number of leaves specified by the user. If N is the number of leaf nodes to show in the taxonomic tree, the function carries out the following:

1. Build a list of leaf nodes (i.e., nodes with no children).
2. Sort leaf node list by summed read count.
3. Mark first N leaf nodes as "keep" and add to a keep list.
4. Add rest of leaf nodes to remove list.
5. For each node in the keep list:
    1. Recurse up tree to root, marking each successive parent as "keep".
6. For each node in the remove list:
    1. Recurse up the tree and find first ancestor marked as "keep".
    2. Hide branch from first "keep" ancestor down with the hideSpecificBranch() function.

3.9.4 Reset tree

The resetTreeBranches() function resets the tree branches and prepares it for filtering and pruning by expanding any click-collapsed nodes and restoring any branches that had been pruned as follows:

1. Recurse down tree from root towards leaves. For each node:
    1. Set “keep” property to false.
    2. If node has been click-collapsed:
        1. Unhide children.
    3. If node has pruned branches:
        1. Add pruned child nodes to property of visible children.
        2. Remove property that stored pruned children.

3.10 Summary and future work

MARTi addresses the need for an open-source software tool that enables real-time analysis and visualisation of metagenomic sequencing data. The tool consists of two main parts, the Engine and GUI, and can be configured in many ways to suit the needs of the user. In local configuration, both the back and front end are run on the same device, typically a desktop or laptop, without the requirement of an internet connection. Combined with a minimalist in-field laboratory setup, MARTi makes it possible to carry out real-time analysis at the site of sample collection. With HPC configuration, analysis processes can be parallelised to a greater extent making it less likely for analysis to fall behind sequencing, enabling larger databases and allowing multiple real-time runs to be analysed simultaneously.

The MARTi GUI features two main analysis modes: Dashboard, for analysing a single sample; and Compare, which allows multiple samples to be viewed side by side. Currently, the majority of plots available on the GUI are for exploring the taxonomic composition of samples, with the AMR table and AMR walkout analysis donut plot being the only forms of functional analysis. One area of future development for MARTi will be to expand its functional analysis offering, mapping reads to databases of gene groups such as KEGG to identify genes with known and annotated functions.

Other areas of future developments for MARTi include: improvements to existing plots, such as added options and features; addition of new plots, such as a treemap plot (Shneiderman 1992), comparison tree, comparison heatmap, and functional analysis plots; developing support for alternative classification methods to the current BLAST LCA pipeline such as Centrifuge, allowing taxonomic analysis to be carried out more quickly and on devices with less memory; addition of multivariate analyses, such as principal component analysis and principal coordinates analysis; and addition of richness and evenness indices, such as the Shannon–Wiener index and Simpson ’s index.

Although MARTi was primarily developed for metagenomic data, the tool could also be used with metabarcoding data such as 16S rRNA for prokaryotes, 18S rRNA for microbial eukaryotes, and Cytochrome Oxidase (CO1) for animals. Barcode databases represent a greater number of species than databases of complete genomes and therefore could reveal a more complete picture of the diversity in a sample. Barcode databases are much smaller than those containing whole references, meaning MARTi Engine could make classifications at a much greater rate. However, the short metabarcoding reads have reduced discriminatory power and lower taxonomic resolution. Furthermore, metabarcoding is less suitable for quantifying relative abundances, due to PCR amplification biases and varying copy numbers of barcode loci, and functional analysis is not possible.

This chapter provided an overview of the MARTi tool and described the structure and methods behind the front end. The next chapter will focus on the application of MARTi, demonstrating the tool in action on a pre-sequenced mock microbial mix and then in real-time on clinical faecal samples.

4 Chapter 4 – Demonstration of MARTi on mock and clinical metagenomic samples

4.1 Introduction

Chapter 3 provided an overview of the MARTi tool and its implementation, including descriptions of key analysis and visualisation algorithms. In this chapter, the results from using MARTi to analyse real experiments are reported – firstly, using a mock gut community and, secondly, using clinical faecal gut microbiome samples taken from both patients suffering from liver disease and healthy individuals.

As metagenomic sequencing studies can suffer from bias and errors at every step of the workflow, from DNA extraction to computational analysis, evaluation on reference communities with known compositions is critical. The ZymoBIOMICS Gut Microbiome Standard (Zymo Research, Irvine, USA), a cell-level pool of 21 microbial strains (18 bacterial, 2 fungal, and 1 archaeal) in staggered abundances was chosen for the first demonstration of MARTi. The mock was designed to mimic a human gut microbiome and provides multiple challenges for metagenomic pipelines, such as difficult-to-lyse Gram-positive bacteria (e.g., *Roseburia hominis*) for testing lysis efficiency, low-abundance organisms for assessing detection limit, and five different strains of *E. coli* for testing taxonomic resolution.

Following the gut mock analysis, MARTi was run on clinical faecal samples from patients with advanced liver cirrhosis (severe scarring of the liver) and from healthy individuals for comparison. Samples from the cirrhosis patients can be divided into three categories:

1. Decompensated cirrhotic - from patients that have developed at least one major complication including ascites, infection, gastrointestinal haemorrhage, and hepatic encephalopathy.

2. Acute-on-chronic liver failure (ACLF) – characterised by sudden worsening of liver function in patients with chronic liver disease and is associated with one or more organ failures and increased mortality.
3. Stable cirrhotic – decompensated cirrhosis patients that rarely require hospital admission and have a much lower mortality risk.

Advanced liver cirrhosis is associated with compositional changes of the gut microbiota, known as dysbiosis, that results in higher levels of pathogens and higher abundances of AMR genes, which are related to an increase in hospitalisations and death independent of cirrhosis severity (Arroyo et al. 2016; Shamsaddini et al. 2021). Identification of AMR genes in the gut microbiome of cirrhosis patients could be used for predicting outcomes and targeting them for therapy (Shamsaddini et al. 2021). Therefore, these samples represent an ideal biological test case for the full range of MARTi functionality. MARTi was used to analyse data from three live sequencing runs:

1. Clinical F3 M – A single sample, F3, that originated from a patient with decompensated cirrhosis was sequenced on a MinION Mk1C. MARTi was run in local configuration, with the Engine and GUI running on a laptop, accessing the MinKNOW pass reads from the mapped drive of the MinION Mk1C.
2. Clinical F3 G – Same sample as previous run, F3, but with this run sequencing was carried out on a GridION and MARTi was run in the HPC configuration.
3. Clinical pool – A pool of 12 faecal samples from patients with cirrhosis, one stable cirrhotic, five decompensated cirrhotic and six ACLF. The barcoded library was sequenced on a GridION and MARTi was run in HPC configuration.

All experimental work in this chapter was performed by the author, except for the DNA extractions and two of the nanopore libraries. The Zymo gut mock DNA extraction nanopore libraries for the *Clinical pool* and *Zymo gut mock* were performed by Darren Heavens, and the clinical faecal sample DNA extractions were carried out by Raymond Kiu, based at the Quadram Institute.

4.2 Methods

4.2.1 Mock gut microbiome DNA extraction

DNA was extracted from the ZymoBIOMICS Gut Microbiome Standard (Zymo Research, Irvine, USA) using an adapted version of the manufacturer's protocol for the Zymo Research Quick-DNA HMW Magbead Kit (Zymo Research). A 200 µl aliquot of the Zymo gut mock cell suspension was centrifuged at 5,000 x g for 5 min at 4 °C. The supernatant was removed and stored in a new 1.5 ml tube. The cell pellet was resuspended in 200 µl of PBS and 10 µl of metapolyzyme solution (10 mg/ml in PBS) then incubated for 2 h at 35 °C in an Eppendorf Thermomixer C (Eppendorf, Hamburg, Germany) operating at 1,000 rpm. The 200 µl of retained supernatant, 20 µl 10% SDS, and 20 µl of Proteinase K were added to the mixture before another incubation at 55 °C for 30 min at 1,000 rpm. The tube was spun at 5,000 x g for 5 min at 4 °C and then 400 µl of the supernatant was transferred to a new 1.5 ml tube.

To precipitate the DNA onto beads, 800 µl of Quick-DNA MagBinding buffer and 50µl of MagBinding beads were added to the tube before a 10 min incubation at room temperature with gentle shaking at 600 rpm. The beads were pelleted on a Dynabeads Magnetic Particle Concentrator (MPC, Thermo Fisher Scientific, Waltham, USA), the supernatant was discarded and then 500 µl of MagBinding buffer was added. The tube was incubated for 10 min at room temperature with mixing at 1,000 rpm. The beads were pelleted on an MPC, the supernatant discarded and then

900 µl of DNA Pre-wash buffer was added. After resuspension of the beads, the tube was returned to the MPC until the beads separated from the solution, then the supernatant discarded. The beads were washed twice using the following protocol: 900 µl of g-DNA Wash buffer was added, the beads were gently resuspended in the buffer by pipetting with a wide-bore tip, the bead solution was transferred to a new 1.5 ml tube, the tube was placed on an MPC until the beads separated from solution, and then the supernatant was discarded. A single 900 µl elution buffer wash was performed whilst the beads remained on the MPC, removing the buffer gently immediately after it was added. The beads were resuspended in 75µl of elution buffer and incubated for 10 min at room temperature with shaking at 400 rpm. The sample was returned to the MPC for a final time until the beads separated from solution, then the eluted DNA was transferred to a new tube. The concentration of the eluted DNA was determined using the dsDNA BR assay for Qubit (Thermo Fisher Scientific).

4.2.2 DNA extraction from clinical and healthy faecal samples

The FastDNA SPIN Kit for Soil (MP Biomedicals, Irvine, USA) was used to extract DNA from ~50-100 mg of faeces per sample following the manufacturer's instructions with one exception, bead-beating was performed twice for a total time of 80s (40s x 2) on a FastPrep 24 (MP Biomedicals) homogeniser.

The concentration of the eluted DNA was assessed on a Qubit fluorometer with the dsDNA BR assay kit. DNA fragment size distributions were checked using the Genomic DNA Analysis ScreenTape on an Agilent TapeStation 2200 (Agilent, Santa Clara, USA).

4.2.3 Library preparation and sequencing of mock gut microbiome

A nanopore sequencing library was prepared from the ZymoBIOMICS Gut Microbiome Standard DNA extract using the SQK-LSK109 kit (Oxford Nanopore Technologies, Oxford, UK) according to the GDE_9063_v109_revAB_14Aug2019 version of the manufacturer's protocol. Sequencing was performed on a GridION using a FLO-MIN106D flow cell and the MinKNOW (v21.05.12) software's standard 72 hour run script. Reads were basecalled live by the Guppy v5.0.12 GPU basecaller with the High accuracy (HAC) model and a minimum pass read Q score of 9.

4.2.4 Library preparation and sequencing of clinical faecal samples

A pooled library of 12 clinical faecal samples was generated with the SQK-LSK109 kit and Native Barcoding Expansion kit (EXP-NBD104) according to the manufacturer's instructions (NBE_9065_v109_revAD_14Aug2019) except that input DNA varied between 40 and 400 ng as some samples contained low amounts of extracted DNA. For pooling ahead of adapter ligation, up to 60 ng of each barcoded sample was combined in a total volume of 65 µl. The pooled library was sequenced on a GridION using a FLO-MIN106D flow cell with the standard 72 hour run script (MinKNOW v21.05.12). Reads were basecalled on-instrument by Guppy (v5.0.12) with the High Accuracy (HAC) model and a minimum pass read Q score of 9.

The individual sample 3 (F3) library was generated with the SQK-LSK109 kit according to the GDE_9063_v109_revAB_14Aug2019 version of the manufacturer's protocol. Two FLO-MIN106D flow cells were loaded with ~30 fmol of the prepared library. One of them was sequenced on a MinION Mk1C for 72 hours with live basecalling using the Fast basecalling model and a minimum pass read Q score of 8. The other flow cell was sequenced on a GridION for the same length of time with

HAC basecalling and a higher minimum pass Q score of 9. Both platforms were using the same versions of MinKNOW (v21.05.12) and Guppy (v5.0.12).

4.2.5 Library preparation and sequencing of healthy control faecal samples

A library of 12 healthy control faecal samples was generated with the Ligation Sequencing kit (SQK-LSK109) and Native Barcoding Expansion kit (EXP-NBD104) according to the manufacturer's instructions (NBE_9065_v109_revAD_14Aug2019). The pooled library was sequenced on a GridION using a FLO-MIN106D flow cell with the standard 72 hour run script (MinKNOW v21.11.7). Reads were basecalled on-instrument by Guppy (v5.1.13) with the High Accuracy (HAC) model and a minimum pass read Q score of 9.

4.2.6 Taxonomic assignment of mock gut microbiome data

The MinKNOW pass read data from the Zymo gut mock sequencing run were analysed by the MARTi Engine (v0.9.2). Reads passing the Engine's default prefilter, minimum length of 500 bp and minimum Qscore of 9, were aligned to sequences of the nucleotide database (nt) using BLAST's megablast algorithm. The Engine assigned reads to taxa using MARTi's LCA algorithm with the default parameters set. Taxonomic assignment results were explored and analysed with the MARTi GUI.

4.2.7 Assessing the effect of chunk size on MARTi analysis rate

To quantify the effect of chunk size on analysis speed, the first 80k pass reads from the Zymo gut mock sequencing run were analysed by the MARTi Engine (v0.9.2) in local configuration on a MacBook Pro (2021 2.3Ghz i9 8-core with 64GB of DDR4 memory) with chunk sizes ranging from 1,000 to 10,000 reads. Reads passing the

Engine's default prefilter were aligned to a database of prokaryotic RefSeq genomes using BLAST's megablast algorithm. For each chunk size, four BLAST jobs were run in parallel, each given four threads.

4.2.8 MARTi analysis of multiplexed clinical samples

Reads of the barcoded clinical faecal samples were demultiplexed in real time by Guppy (v5.0.12) running as part of MinKNOW (v21.05.12) on a GridION. The pass reads of each sample were analysed live by the MARTi Engine during the sequencing run. The Engine aligned reads to prokaryotic RefSeq genomes, using the megablast algorithm, for taxonomic assignments and the CARD database, with blastn, for AMR analysis. The MARTi GUI was used to view and explore the taxonomic compositions and AMR results of the clinical samples.

4.2.9 MARTi analysis of the healthy control pool

Demultiplexed pass reads of each sample were analysed by the MARTi Engine post sequencing. The Engine aligned reads to prokaryotic RefSeq genomes, using the megablast algorithm, for taxonomic assignments and the CARD database, with blastn, for AMR analysis. The MARTi GUI was used to view and explore the taxonomic compositions of the healthy control samples and compare them to the previously sequenced clinical samples.

4.3 Results

4.3.1 Sequencing metrics

A total of 98.7 gigabases (Gb) was generated from the five nanopore flow cells (Table 4.1). The number of reads produced per flow cell ranged from 1.3 M for the gut mock run to 8.2 M for the pool of 12 clinical faecal samples. The gut mock run had the highest read length N50, 19.57 kb, for reads passing the basecaller's quality filter, whilst the pass read N50's of the four faecal sample runs ranged from 5.09 to 6.40 kb. The mean pass read Q scores were higher for the runs basecalled with the HAC model, 20.11 – 20.93, in comparison to the run basecalled with the Fast model, 19.15, which also had the highest percentage of pass reads, 78.3%

Table 4.1 Summary of data obtained from mock and clinical metagenomic nanopore sequencing runs

Run	Clinical sample ID	Barcode	Sequencer	Yield (Gb)	Reads (M)	Qscore filter	Basecall model	Pass yield (Gb)	Pass reads (M)	Pass N50 (kb)	Pass Quality (mean base Q)
Zymo gut mock	-	-	GridION	12.79	1.30	9	HAC	9.74	0.94	19.57	20.11
Clinical F3 M	F3	-	MinION Mk1C	24.70	6.55	8	Fast	20.06	5.12	5.67	19.15
Clinical F3 G	F3	-	GridION	23.51	5.32	9	HAC	18.00	3.91	6.40	20.80
Clinical pool	Pool	Combined	GridION	20.58	8.24	9	HAC	16.11	6.19	5.46	20.48
	F6	01						2.47	0.84	5.84	20.16
	F17	02						1.95	0.54	6.61	20.76
	F18	03						1.58	0.62	5.81	20.22
	F40	04						1.50	0.30	6.98	21.08
	F52B	05						1.14	0.25	6.81	20.60
	F57	06						0.74	0.39	4.18	21.11
	F58	07						1.41	0.61	4.81	20.44
	F82	08						1.32	0.61	3.86	20.30
	F129	09						0.07	0.02	5.87	20.58
	F137	10						1.12	0.70	3.10	20.13
	F139	11						0.47	0.26	4.50	20.84
	F140	12						0.62	0.36	3.70	20.23
	-	Unclassified						1.73	0.69	5.36	20.37

Table 4.1 continued…

Run	Clinical sample ID	Barcode	Sequencer	Yield (Gb)	Reads (M)	Qscore filter	Basecall model	Pass yield (Gb)	Pass reads (M)	Pass N50 (kb)	Pass Quality (mean base Q)
Healthy pool	Pool	Combined	GridION	17.12	7.62	9	HAC	12.05	5.41	5.09	20.93
	HC2R	01						0.89	0.51	4.32	21.01
	HC76	02						0.78	0.36	4.95	20.97
	HC77	03						0.92	0.35	5.22	21.00
	HC79	04						1.14	0.34	5.84	21.06
	HC80	05						1.19	0.31	5.79	20.98
	HC89	06						1.11	0.53	4.52	20.86
	HC86	07						0.88	0.30	6.14	21.00
	HC65	08						0.74	0.47	4.30	20.86
	HC61	09						1.00	0.50	5.21	20.96
	HC59	10						0.81	0.42	4.95	20.96
	HC44	11						1.00	0.56	4.36	21.00
	HC51	12						1.03	0.55	3.92	21.00
	-	Unclassified						0.57	0.21	5.68	20.10

4.3.2 Expected composition of mock mix vs MARTi assignments

The pass reads from the Zymo gut mock run were analysed by the MARTi Engine. In total, 905,235 (95.9%) of the Zymo gut mock pass reads made it through MARTi's default prefiltering. Of those passing filter, 882,571 (97.5%) could be assigned to a taxon by the Engine's BLAST-LCA pipeline.

With no minimum abundance cutoff set for the LCA algorithm, 86.5% of reads assigned at the species level were assigned to species known to be present in the mock (Table 4.2). Only one out of the 17 species in the mock, *Veillonella rogosae*, was unable to be detected at species level. Most (90.3%) of the reads assigned to "Other" species were assigned to congeners of species present in the mock, with the majority being assigned to members of the *Veillonella* (56.0%) and *Prevotella* (30.1%) genera. Assigned read proportions to species present in the mock community positively correlate with the theoretical abundances (Figure 4.1a, log-transformed Pearson's r = 0.66).

Table 4.2 Summary of reads assigned by MARTi at the species level for the Zymo gut mock run

Species	Assigned reads	Theoretical abundance (%)	MARTi proportions (%)
Bacteroides fragilis	103,386	14	13.55
Escherichia coli	86,681	14	11.36
Faecalibacterium prausnitzii	207,965	14	27.26
Roseburia hominis	58,161	14	7.63
Veillonella rogosae	0	14	0.00
Bifidobacterium adolescentis	1,048	6	0.14
Fusobacterium nucleatum	46,918	6	6.15
Lactobacillus fermentum	110,609	6	14.50
Prevotella corporis	294	6	0.04
Akkermansia muciniphila	17,164	1.5	2.25
Candida albicans	5,158	1.5	0.68
Clostridioides difficile	17,364	1.5	2.28
Saccharomyces cerevisiae	4,386	1.4	0.58
Methanobrevibacter smithii	189	0.1	0.02
Salmonella enterica	136	0.01	0.02
Enterococcus faecalis	5	0.001	0.001
Clostridium perfringens	2	0.0001	0.0003
Other	103,296	0	13.54

All of the reads classified at the genus level using a 0.1% minimum abundance cutoff for the LCA algorithm were assigned to genera present in the gut mock with no false-positive genera detected (Table 4.3). Furthermore, all of the gut mock genera with theoretical abundances greater than the 0.1% cutoff (13 out of the 17 genera) were detected. The proportions of reads classified by MARTi at the genus level correlate with the expected proportions of genera in the mock (Figure 4.1b, log-transformed Pearson’s r = 0.77).

Table 4.3 Summary of reads assigned by MARTi with a 0.1% minimum abundance cutoff at the genus level for the Zymo gut mock run

Genus	Assigned reads	Theoretical abundance (%)	MARTi proportions (%)
Bacteroides	105,952	14	13.02
Escherichia	86,832	14	10.67
Faecalibacterium	208,136	14	25.58
Roseburia	59,403	14	7.30
Veillonella	92,526	14	11.37
Bifidobacterium	1,087	6	0.13
Fusobacterium	50,440	6	6.20
Lactobacillus	125,895	6	15.47
Prevotella	39,300	6	4.83
Akkermansia	17,164	1.5	2.11
Candida	5,161	1.5	0.63
Clostridioides	17,364	1.5	2.13
Saccharomyces	4,408	1.4	0.54
Methanobrevibacter	0	0.1	0.0
Salmonella	0	0.01	0.00
Enterococcus	0	0.001	0.000
Clostridium	0	0.0001	0.0000

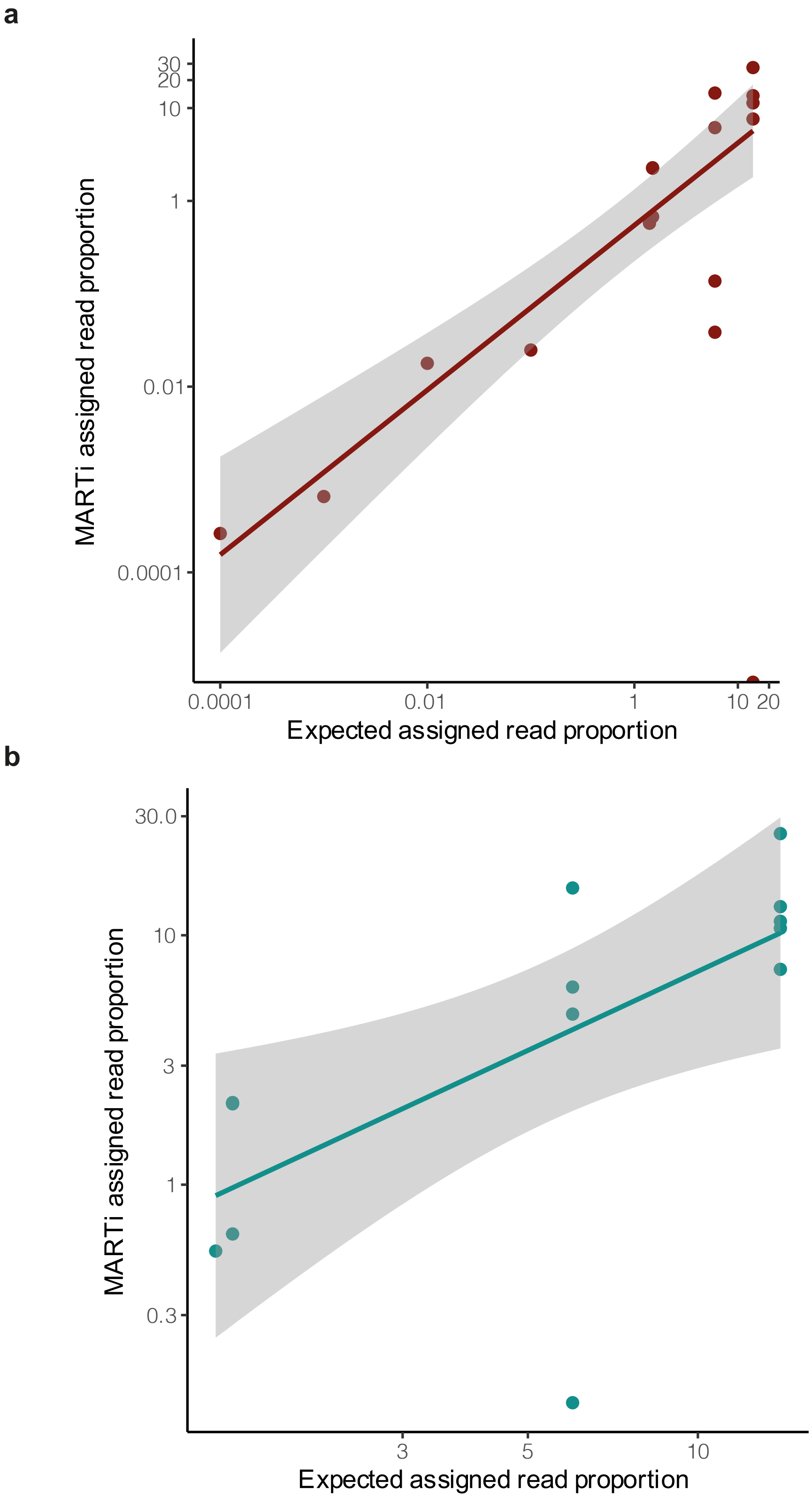


Figure 4.1 Correlation plots comparing the expected proportions of Zymo gut mock community taxa with proportions obtained from MARTi analysis of nanopore sequencing reads. ***a*** *– Correlation at species level with no LCA minimum abundance cutoff (log-transformed Pearson's r = 0.66).* ***b*** *– Genus-level correlation with a minimum abundance threshold of 0.1% (log-transformed Pearson's r = 0.77). The grey region either side of the fit line represents the 95% Confidence Intervals.*

4.3.3 The effect of chunk size on MARTi analysis rate

To quantify the effect of chunk size on analysis speed, the first 80k pass reads from the Zymo gut mock sequencing run were analysed by the MARTi Engine in local configuration with chunk sizes ranging from 1,000 to 10,000 reads (Figure 4.2). The mean time to complete the first four BLAST jobs gives an indication of how long the GUI would be waiting to display the first results. The mean time to first result ranged from ~10.5 min for 4x1,000 read chunks to ~99 min for 4x10,000 read chunks, with average time per chunk increasing linearly for chunk sizes in between (Figure 4.2). Analysis of all 80k reads completed in a similar time for all chunk sizes, with a mean time of ~3.19 h (Figure 4.2). Over the course of analysis, parallel jobs become out of sync meaning that GUI updates happen more frequently. The mean analysis time per chunk gives an insight into how often the GUI would update taking this into account. With this particular local configuration, mean time between GUI updates ranged from ~2.5 min, for 1,000 read chunks, to ~23.25 for 10,000 reads (Figure 4.2).

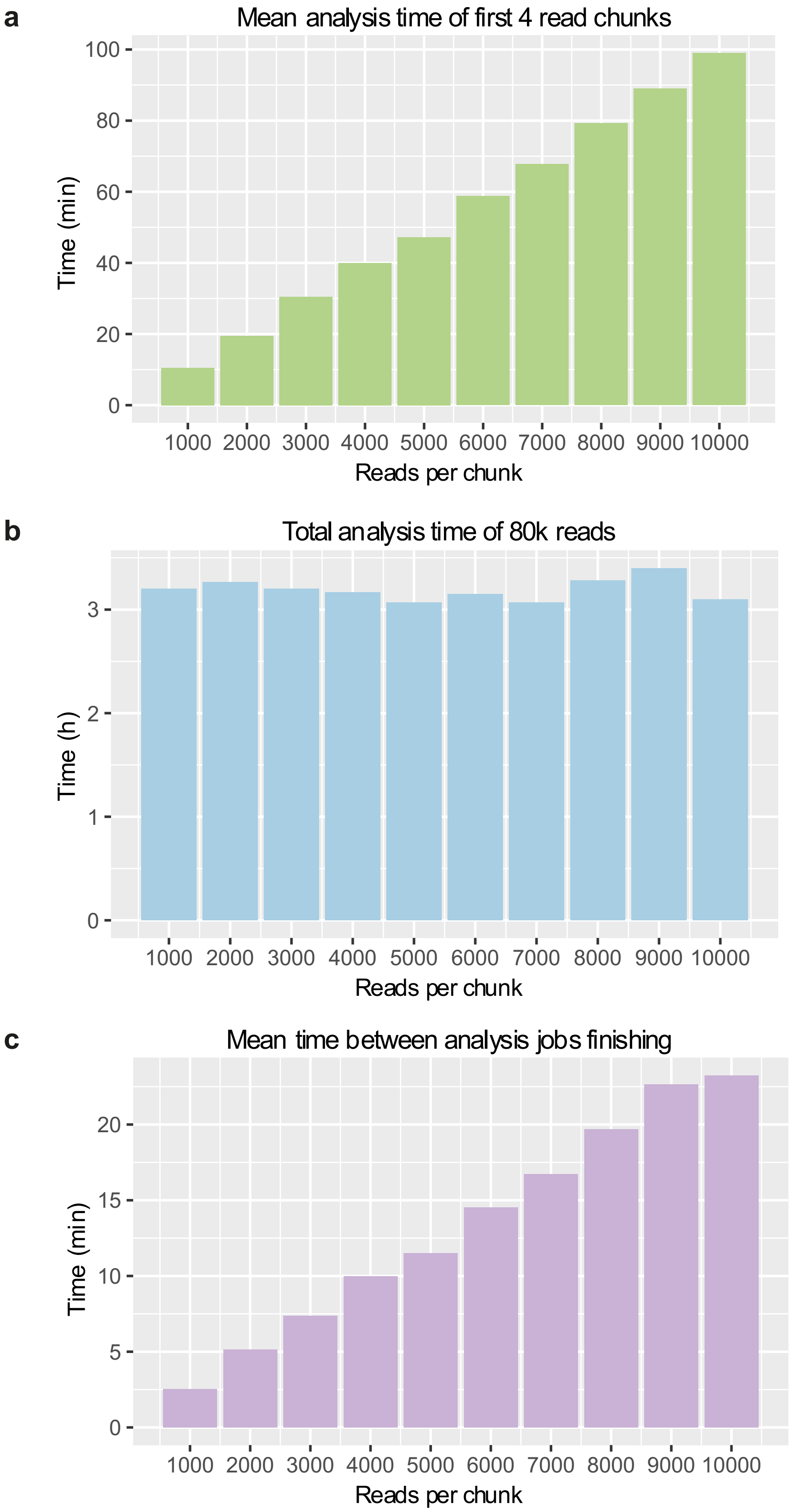
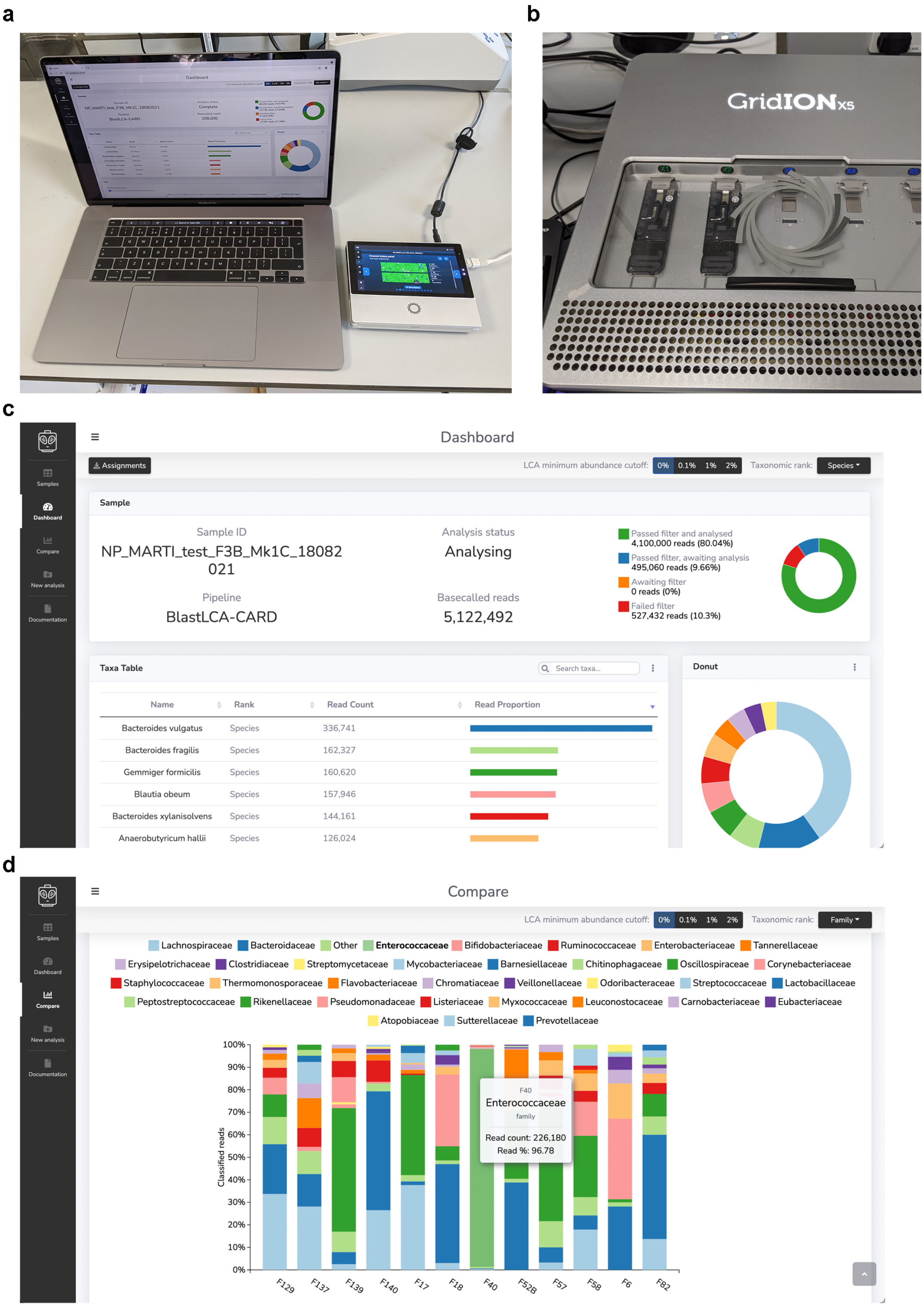


Figure 4.2 MARTi analysis rates using different read chunk sizes, ranging from 1k to 10k. Running in local configuration, MARTi analysed the first 80k gut mock microbiome reads. ***a*** *– Mean time, in minutes, to analyse first four parallel read chunks.* ***b*** *– Total time, in hours, to analyse 80k reads.* ***c*** *– Mean wait, in minutes, between read chunks finishing.*

4.3.4 Real-time analysis of clinical samples

The different configurations of MARTi were tested in real time during three sequencing runs. For the first run, a single clinical faecal sample, F3, was sequenced on a MinION Mk1C (Figure 4.3a). MARTi was configured for local analysis, with the Engine and GUI running on the same laptop. Basecalled pass reads were accessed via the mapped MinION drive. The second run was loaded with the same clinical sample, F3, but was sequenced on a GridION and MARTi was set up in the HPC configuration (Figure 4.3b). The third run was also sequenced on a GridION with real-time HPC analysis, but this run had 12 faecal samples multiplexed onto a single flow cell. This was to test the ability of MARTi to process multiplexed samples.

The MARTi GUI was successfully used to explore the taxonomic composition and AMR gene content of the clinical samples in real time. The Dashboard page enabled real-time monitoring of individual samples (Figure 4.3c) and the Compare page was used to view the taxonomic compositions of the multiplexed clinical samples side-by-side (Figure 4.3d).



Dashboard

Assignments

LCA minimum abundance cutoff: 0% 0.1% 1% 2%

Taxonomic rank: Species

Sample

Sample ID
NP_MARTI_test_F3B_Mk1C_18082021

Analysis status
Analysing

Pipeline
BlastLCA-CARD

Basecalled reads
5,122,492

🟩 Passed filter and analysed
4,100,000 reads (80.04%)

🟦 Passed filter, awaiting analysis
495,060 reads (9.66%)

🟧 Awaiting filter
0 reads (0%)

🟥 Failed filter
527,432 reads (10.3%)

Taxa Table

Search taxa...

Name	Rank	Read Count	Read Proportion
Bacteroides vulgatus	Species	336,741	
Bacteroides fragilis	Species	162,327	
Gemmiger formicilis	Species	160,620	
Blautia obeum	Species	157,946	
Bacteroides xylanisolvens	Species	144,161	
Anaerobutyricum hallii	Species	126,024	

Donut

Compare

LCA minimum abundance cutoff: 0% 0.1% 1% 2%

Taxonomic rank: Family

F40
Enterococcaceae
family
Read count: 226,180
Read %: 96.78

Figure 4.3 Real-time analysis of clinical samples using MARTi. **a** – The local configuration of MARTi, with the Engine and GUI running on a single laptop, was tested on a live MinION Mk1C sequencing run of a clinical faecal sample (F3). **b** – MARTi analysed two GridION runs, a single clinical sample (F3) and a pool of clinical samples, in real time running in the HPC configuration. **c** – A screenshot of the Dashboard page of the MARTi GUI during live analysis of a clinical faecal sample. **d** – The Compare page during real-time metagenomic analysis of a pool of 12 faecal samples.

4.3.5 MARTi HPC configuration analysis rate during a live sequencing run

The *Clinical F3 G* run on the GridION produced 2.24 million basecalled pass reads in the first 24 hours of sequencing, at an average rate of 1,556 pass reads per minute (Figure 4.4). Analysing in real-time on an HPC, MARTi BLASTed batches of filtered reads in chunks of 4,000 to prokaryotic RefSeq genomes for taxonomic classification and CARD for AMR identification. Up to six chunks of reads were analysed in parallel. Within 24 hours, MARTi had analysed just over 1.83 million reads at an average rate of 1,273 reads per minute.

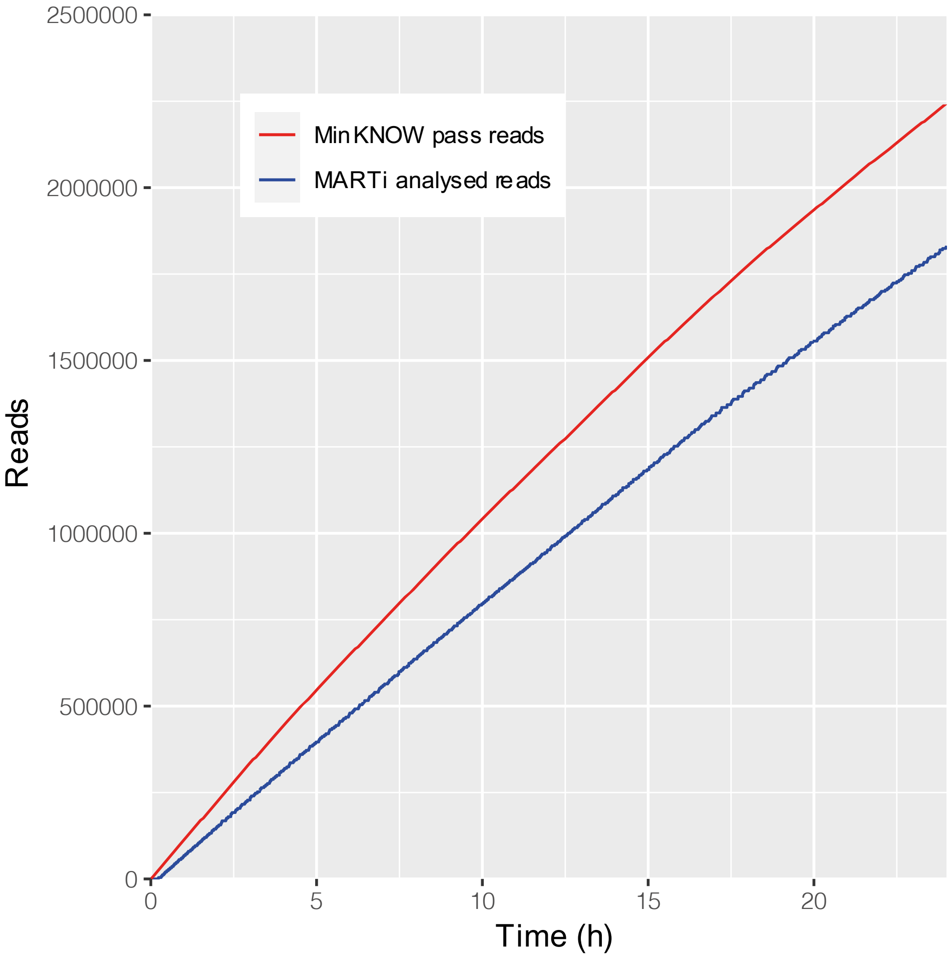


Figure 4.4 Basecalled pass read production from a clinical faecal sample on the GridION (red line) and reads analysed by MARTi running in HPC configuration (blue line) during the first 24 hours of sequencing.

4.3.6 Characterisation of barcoded clinical samples

The *Clinical pool* run generated a total of 20.6 gigabases of data with 6.19 million reads passing MinKNOW's minimum quality score filter (Table 4.1). Reads were demultiplexed in real time and 88.9% were successfully assigned to a barcode. The pass reads from each of the barcoded samples were analysed live by the MARTi Engine during the sequencing run. MARTi GUI's Compare page was used to view the taxonomic compositions of the 12 clinical samples side-by-side. At the phylum level, it was observed that the majority of the samples (11/12) contained the three bacterial phyla that are usually dominant in stable adult microbiota: *Firmicutes*, *Bacteroidetes*, and *Actinobacteria* (Figure 4.5). At the family level, many of the patients have high levels of *Enterococcaceae*, up to 96.8% for F40 (Figure 4.6). Enterococcus is associated with complications in patients with end-stage liver disease (Llorente et al. 2017).

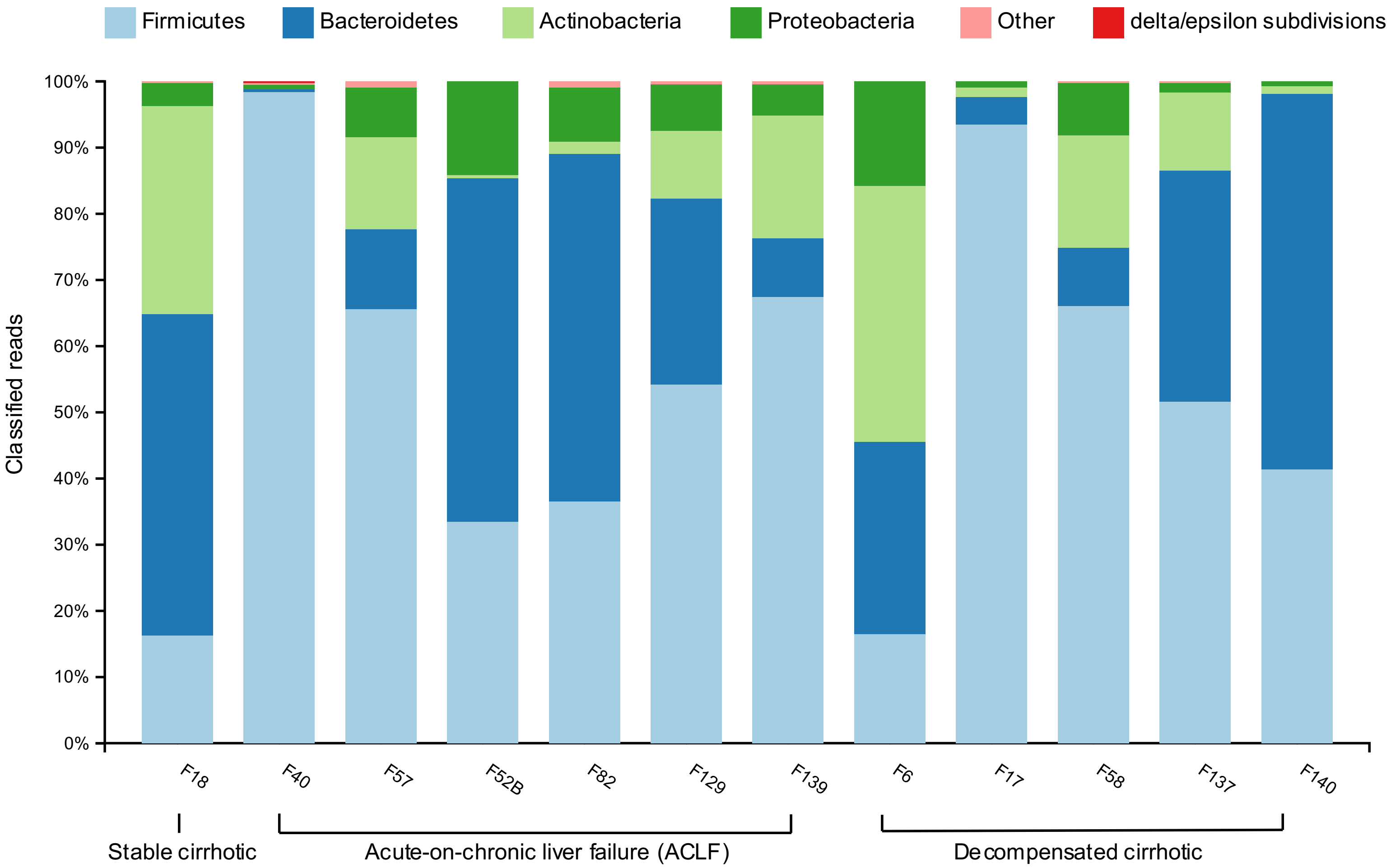


Figure 4.5 Phylum-level composition of clinical faecal samples from patients with liver disease. The barcoded samples were analysed by the MARTi Engine in real-time and the results were monitored and explored using the MARTi GUI. This figure was exported from the stacked bar card on the GUI's Compare page as an SVG and then annotated using Adobe Illustrator.

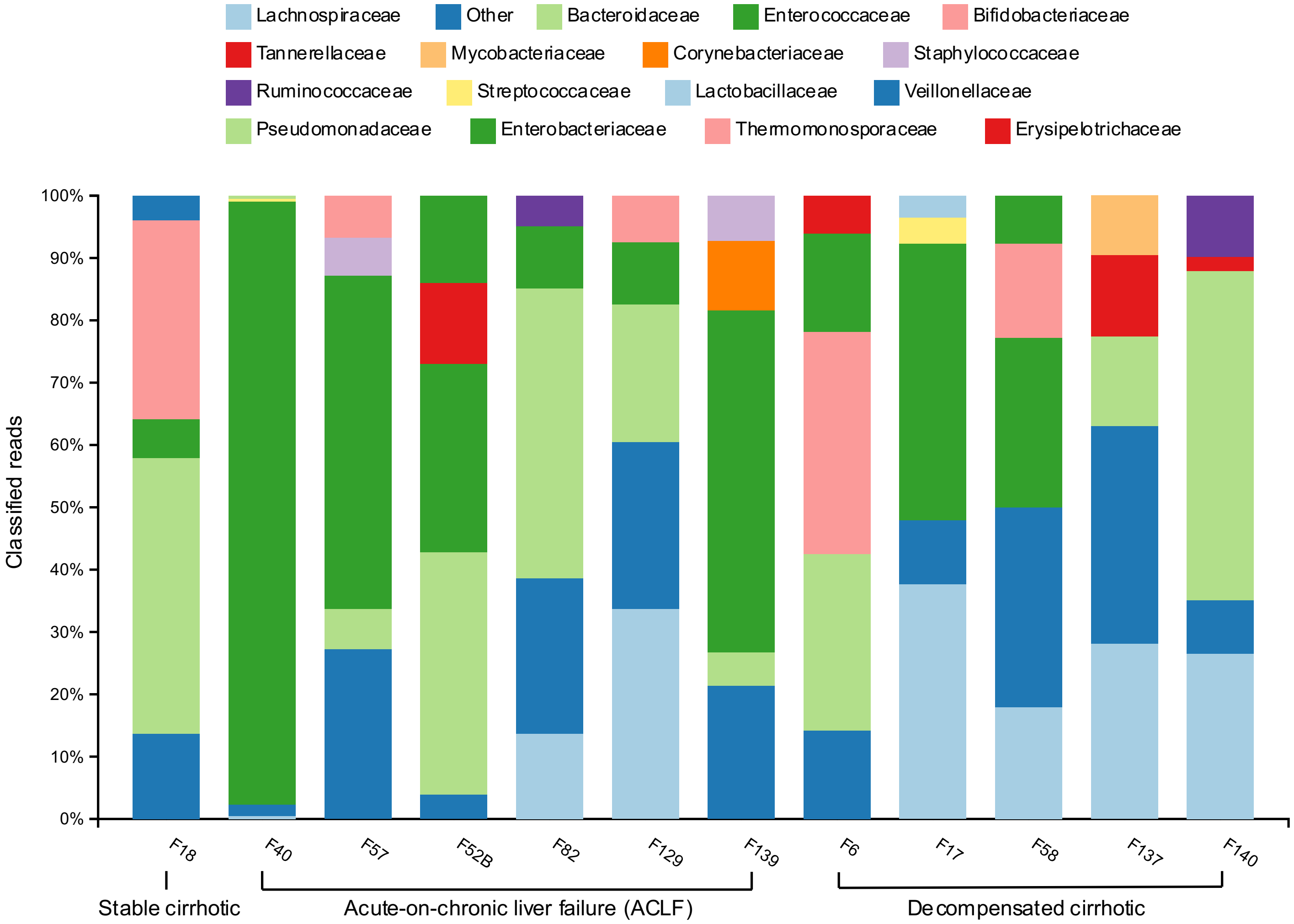


Figure 4.6 Family-level composition of clinical faecal samples from patients with liver disease. The samples were analysed in real time using MARTi. This figure was generated by the Compare page of the GUI and shows the top 4 taxa at family level for each sample. Annotation was added to the MARTi-exported SVG using Adobe Illustrator.

4.3.7 Comparison of F3 taxonomic composition between run configurations

The nanopore sequencing library for clinical faecal sample F3 was loaded onto two MinION flow cells, one of which was sequenced on a GridION and the other on a MinION Mk1C. Both runs had live basecalling enabled. The more computationally demanding HAC basecalling model was used on the GridION and the Fast model was used on the MinION. MARTi was used to analyse the runs in real time, running on the HPC for the GridION experiment and on a laptop for the MinION. At species level, with a 1% minimum abundance cutoff for the LCA algorithm, 15 different species could be detected in the sample on both runs (Table 4.4). The proportions of MARTi-classified reads at species level between the two F3 sequencing runs are very similar and show strong positive correlation (Figure 4.7, Pearson's r = 0.99).

Table 4.4 Summary of reads assigned by MARTi at the species level with a 1% LCA minimum abundance cutoff for each F3 sequencing run.

Species	Clinical (F3) G		Clinical (F3) M	
	Assigned reads	Proportion (%)	Assigned reads	Proportion (%)
Bacteroides vulgatus	297,904	18.31	378,430	19.57
Gemmiger formicilis	163,837	10.07	179,013	9.26
Blautia obeum	159,907	9.83	176,703	9.14
Bacteroides fragilis	143,445	8.82	182,130	9.42
Bacteroides xylanisolvens	127,901	7.86	162,080	8.38
Anaerobutyricum hallii	124,707	7.67	141,273	7.30
Faecalibacterium prausnitzii	107,983	6.64	118,571	6.13
Klebsiella variicola	88,117	5.42	110,764	5.73
Alistipes finegoldii	83,765	5.15	104,915	5.42
Enterococcus faecium	77,539	4.77	84,061	4.35
Barnesiella intestinihominis	70,429	4.33	91,017	4.71
Bifidobacterium longum	54,553	3.35	58,643	3.03
Blautia faecis	45,755	2.81	51,036	2.64
Bacteroides dorei	41,956	2.58	50,973	2.64
Blautia wexlerae	39,012	2.40	44,315	2.29

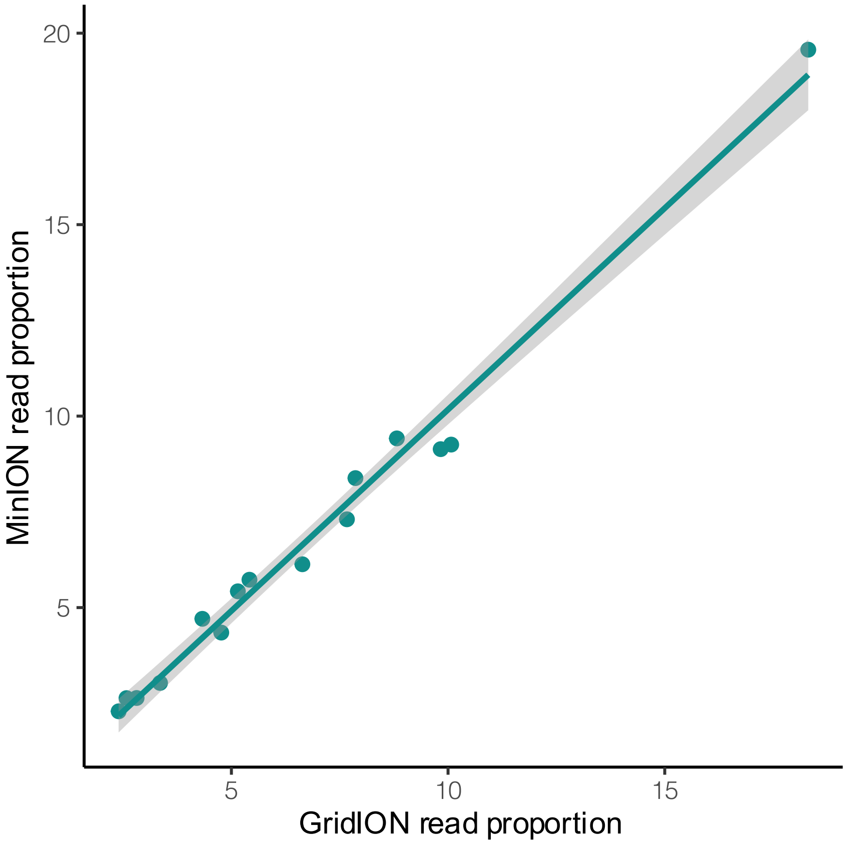


Figure 4.7 Correlation plot of MARTi's species-level read assignments for a clinical faecal sample (F3) sequenced on the GridION and MinION Mk1C (Pearson's r = 0.99). A 1% minimum abundance cutoff was used for MARTi Engine's LCA binning. The grey region either side of the fit line represents the 95% Confidence Intervals.

4.3.8 AMR analysis of clinical faecal samples in real time

The MARTi Engine aligned reads to prokaryotic RefSeq genomes and to the CARD database as they were generated. The Engine also carried out walkout analysis from AMR genes into the flanking DNA to identify host organisms. The MARTi GUI provided a graphical user interface to view ARO (Antibiotic Resistance Ontology) hits and walkout analysis results during the live sequencing run. The number of unique AROs detected in the first 100k reads of each clinical faecal sample ranged from 24 to 104, whilst the total number of ARO hits ranged from 223 to 3321 (Table 4.5). MARTi's walkout analysis assigned 98.5% of the ARO hits to taxonomies, with 41.1% of those being assigned at the species or strain level.

The sample that had the fewest number of unique AROs, F40, also had the highest number of ARO hits. The microbiome of F40 was almost entirely dominated by the *Enterococcaceae* family (Figure 4.6) and the walkout analysis showed its AROs were primarily associated with pathobionts belonging to *Enterococcus*, mostly *Enterococcus faecium* (Figure 4.8). When analysing the first 100k reads, the pathobiont *Enterococcus faecium* was found in the top 10 walkout taxa for 7 of the clinical samples, and in the top 20 taxa for 9 of the 11 samples that had at least 100k reads. However, when all available reads were analysed for each sample, *E. faecium* was present in the top 20 for all samples except F18, the microbiome from the stable cirrhotic patient, and in the top 10 walkout taxa for 8 out of 11 samples.

Table 4.5 Antibiotic Resistance Ontology (ARO) results from the first 100k reads of clinical microbiome samples analysed by MARTi

Sample	Cirrhosis condition	No. unique AROs	Total ARO hits	Proportion of reads (%)
F18	Stable	90	536	0.54
F40	ACLF	24	3321	3.32
F57	ACLF	56	540	0.54
F52B	ACLF	104	3274	3.27
F82	ACLF	101	695	0.70
F139	ACLF	60	1143	1.14
F6	Decompensated	89	937	0.94
F17	Decompensated	30	704	0.70
F58	Decompensated	64	415	0.42
F137	Decompensated	57	223	0.22
F140	Decompensated	39	301	0.30

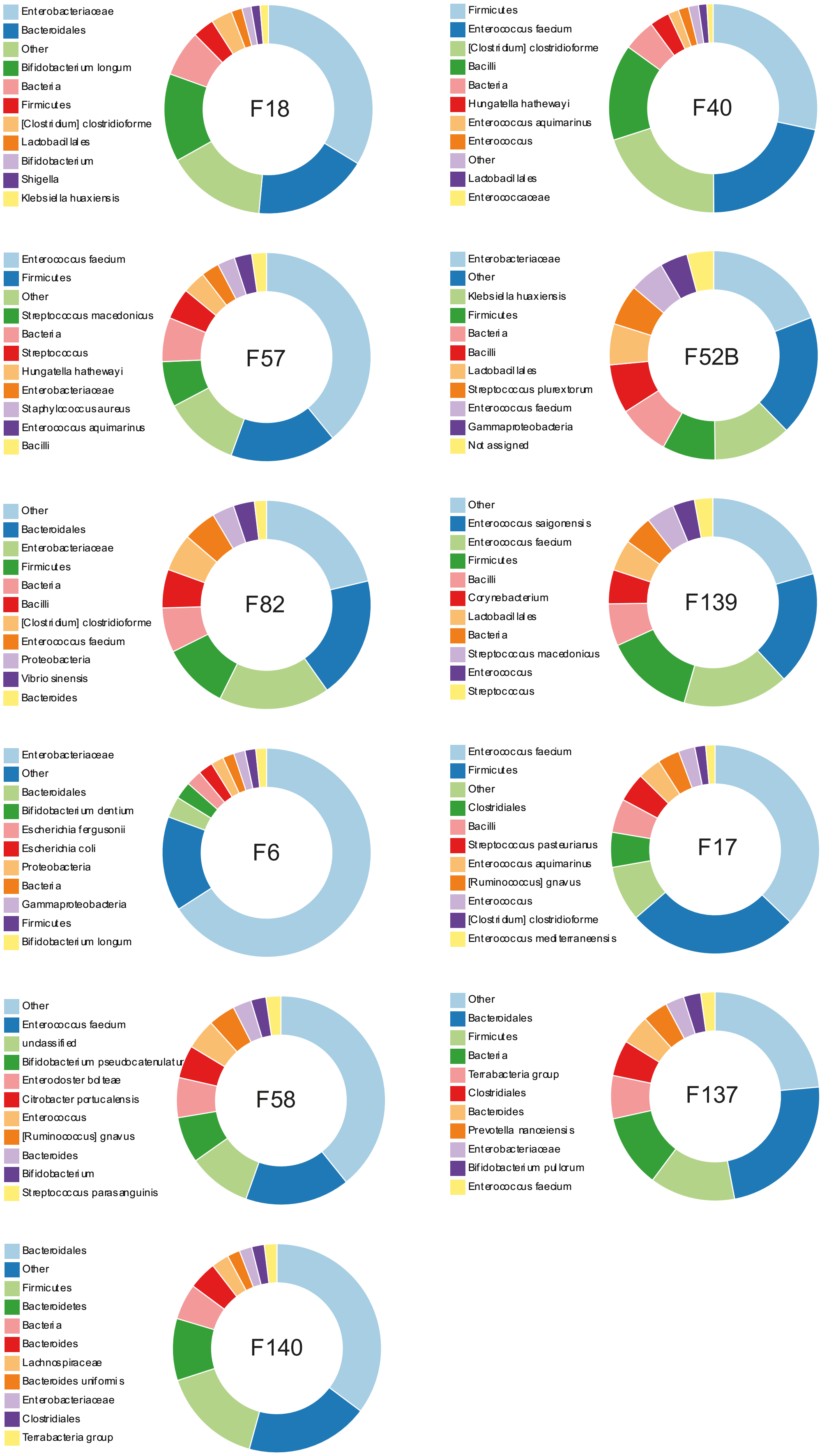


Figure 4.8 AMR walkout analysis donut plots exported from the MARTi GUI. Each donut represents walkout hits from the first 100k reads of clinical faecal samples.

4.3.9 Comparison of gut microbiome species richness between cirrhosis patients and healthy controls

The *Healthy pool* run generated a total of 17.12 gigabases of data with 7.62 million reads passing MinKNOW's minimum quality score filter (Table 4.1). The MARTi Engine was used to analyse the pass reads from each of the barcoded samples and then the MARTi GUI was used to view the taxonomic compositions and species accumulation curves of the 12 healthy sample and the 12 clinical samples (Figure 4.9). An unequal variances t-test was conducted to compare species richness between the healthy and clinical pools after the first 100k reads were analysed by MARTi. There was a significant difference in species richness for the healthy pool (M=1506.75, SD=413.57) and cirrhotic clinical pool (M=334.27, SD=212.57); t=8.653, df=16.726, p<0.001.

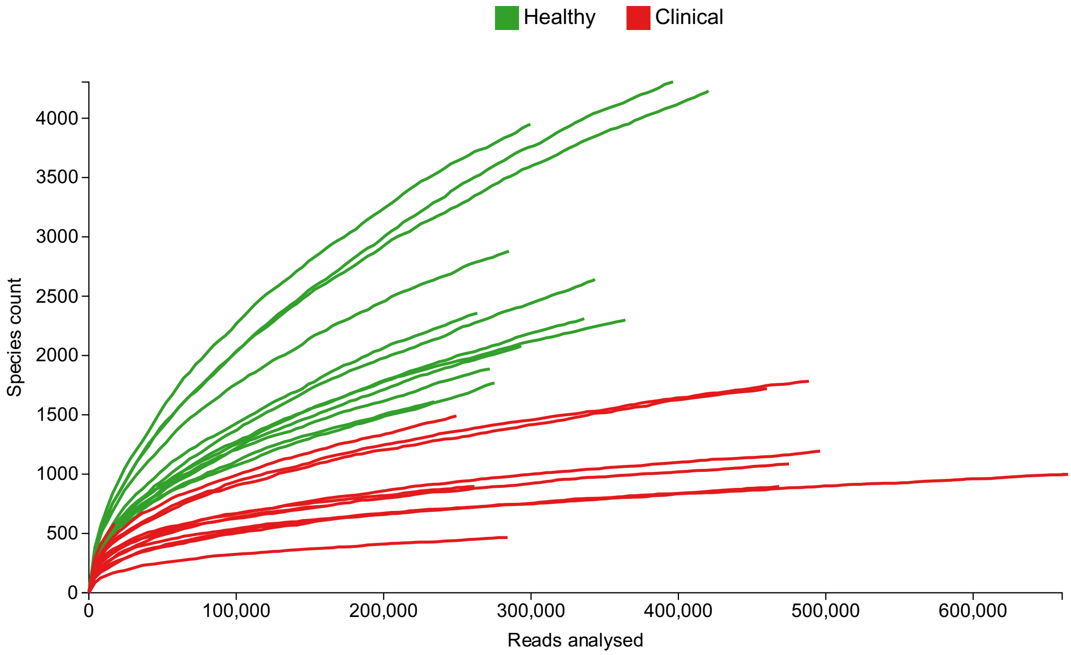


Figure 4.9 Species accumulation plot generated by the MARTi GUI representing all of the samples from the Healthy pool and the Clinical pool. The line colours and legend were altered using Adobe Illustrator.

4.4 Discussion

ONT's long-read sequencing platforms are the first to enable progressive real-time analysis of data and have the potential to revolutionise metagenomics by improving classification accuracy, metagenomic assembly, sequencing efficiency, and by reducing the time to result. However, the full potential of nanopore sequencing remains largely unrealised due to the lack of open source, offline, real-time analysis tools and pipelines. ONT's own EPI2ME platform provides near real-time analysis but has limitations due to its closed nature. Recognising the need for an open, extensible platform, we developed MARTi, an open-source software tool that enables real-time analysis and visualisation of metagenomic sequencing data. MARTi provides a rapid, lightweight web interface that allows users to understand community composition and identify antimicrobial resistance (AMR) genes in real time. In this chapter, we reported on the use of MARTi in laboratory conditions and demonstrated a number of possible configurations of sequencer and software. All of the nanopore runs covered in this chapter were successful on the first attempt and the data generated for the clinical samples will be used as part of a wider study with greater sample numbers.

4.4.1 Testing MARTi on a mock microbiome

We tested MARTi on data from a sequencing run of the ZymoBIOMICS Gut Microbiome Standard, a cell-level pool constructed from quantified pure cultures of 21 microbial strains (18 bacterial, 2 fungal, and 1 archaeal). The mix provides multiple challenges for a metagenomic pipeline, including: the presence of difficult-to-lyse Gram-positive bacteria for testing lysis efficiency; low-abundance organisms, down to a ten-thousandth of a percent, for assessing detection limit; and multiple strains of *E. coli* for testing taxonomic resolution.

Using the default LCA values, MARTi only assigned three reads at the strain level and therefore the five *E. coli* strains could not be distinguished from one another. This is not too surprising as MARTi's default LCA parameter values are tuned towards minimising false positive results rather than low-level taxonomic assignment. For a given read, a set of "good hits" was identified by finding the highest scoring hit (according to the BLAST bitscore), then finding all hits, up to a limit of 20, with a score within 90%. Better strain level resolution could be achieved by opting for a more stringent score percent and by decreasing the number of "good hits" to be considered by the LCA. However, it's likely that this approach would also increase the number of false positive low-level assignments.

MARTi detected almost all of the mock mix species, including the four species present at less than or equal to 0.1%, with just a single species missing (*Veillonella rogosae,* Table 4.2). The missing species can be explained by looking at the reads grouped into the "Other" category. Most (90.3%) of the falsely classified reads were assigned to congeners of species in the mock, with the majority of those assigned to congeners of the two most underrepresented species, *Veillonella* rogosae and *Prevotella corporis*. Although the median nanopore error rates are dropping over time, and are typically 5% or below now, this level of error obscures differences between closely related species such as those from the *Veillonella* and *Prevotella* genera. This also results in some reads being assigned at a higher taxonomic level than species.

A stronger correlation between the proportions of summed read counts and expected mock mix proportions was achieved at the genus level (Figure 4.1). Furthermore, there were no false-positive genera as all of the reads classified at the genus level using a 0.1% minimum abundance cutoff for the LCA algorithm were assigned to genera present in the gut mock (Table 4.3). There were also no false negatives as all of the gut mock genera with theoretical abundances greater than the 0.1% cutoff were detected.

Although a stronger correlation was achieved between the expected and observed summed read counts at genus level compared to species level (Figure 4.1, log-transformed Pearson's r = 0.77 and r = 0.66 respectively), several sources of quantitative error in the method remain. First, DNA extraction efficiency varies across species as some cells are easier to lyse than others. More aggressive extraction methods, such as physical bead beating, could be implemented to achieve better extraction efficiency across the species in the mix, but this would lead to a decrease in DNA fragment length and therefore a decrease in taxonomic assignment resolution. One potential method for reducing DNA extraction bias whilst maintaining the necessary DNA fragment length is the 'Three Peaks' faecal DNA extraction, where the supernatant is kept after each of three successive extraction methods, chemical, enzymatic, and physical, and then pooled before a clean-up step (Quick et al. 2019). This allows more aggressive DNA extraction methods to be used without further fragmenting the DNA of the easier to lyse species.

A second potential source of quantitative error comes from the read length variation between species as this influences the count-based abundances. Average read lengths for species in the mock varied greatly, from 2,090 bp to 46,824 bp. The two most overrepresented species, *Lactobacillus fermentum* (expected 6% abundance, observed 14.5%) and *Faecalibacterium prausnitzii* (expected 14% abundance, observed 27.3%), had the shortest mean read lengths, 2090 bp and 6412 bp respectively. Where read lengths vary greatly between species in a mix, better abundance estimates might be achieved using the sequence yield for each species instead of read count. In common with other metagenomic tools such as MEGAN, all of the taxonomic abundance plots on the MARTi GUI rely on assigned read counts, but with further development future versions could allow the user to switch between read counts and yield.

A third cause is the nanopore error rate. Despite the reduction in error rate over time due to upgrades in hardware, chemistry, and basecalling, the current nanopore error rate (typically < 5%) is still high enough to obscure differences between closely related species. By default, when reads have hits to multiple species in the top 20 hits, which can happen more frequently with higher error rates, MARTi will assign the read to a higher taxonomic level, the lowest common ancestor of those species. When this occurs, the abundance profiles at lower taxonomic levels becomes less accurate. Pearman et al. (2020) observed that recall (defined as the ratio of correctly classified reads to all reads) for long nanopore reads was equal to or higher than the longest Illumina reads (300 bp) and suggested that a simple way to improve classification accuracy of error-prone nanopore reads is to implement minimum read lengths.

4.4.2 The effect of chunk size on MARTi analysis rate

The first 80k pass reads from the gut mock sequencing run were analysed by the MARTi Engine in local configuration with chunk sizes ranging from 1,000 to 10,000 reads (Figure 4.2). The results give potential users an idea of the timings involved for MARTi analysis using different read chunk sizes. On a 2021 MacBook Pro, with four parallel 4-thread BLAST jobs, all chunk sizes had similar rates of analysis with a mean of ~395 reads per minute. The mean time to first result ranged from ~10.5 min for 1,000 read chunks (4,000 reads total) to ~99 min for 10,000 read chunks (40,000 reads), with average time per chunk increasing linearly for chunk sizes in between (Figure 4.2a). The mean time between GUI updates ranged from ~2.5 min for 1,000 read chunks to ~23.25 for 10,000 reads (Figure 4.2c). In a time-critical situation the user should opt for a smaller chunk size as more frequent updates are likely to be more important in terms of time to result than a potentially very slight analysis rate gain from using larger chunk sizes.

4.4.3 Real-time analysis of clinical samples

MARTi was successfully used to analyse the taxonomic composition and AMR gene content of clinical microbiome samples in real time (Figure 4.3). The main configurations of the tool, HPC and Local, were demonstrated on live sequencing runs on the GridION and MinION Mk1C. The MARTi GUI was used to explore the results in real time.

For the *Clinical F3 M* run, a single clinical faecal sample (F3) was sequenced on the MinION Mk1C and MARTi was run in local configuration on the MacBook Pro described in the previous section. The same library was also sequenced on the GridION during the *Clinical F3 G* run, which was analysed by MARTi in the HPC configuration. For both runs, reads were BLASTed to CARD and a database of prokaryotic RefSeq genomes. MARTi running in the HPC configuration was able to analyse reads at a faster rate (1,273 reads per min) than in local configuration (~740 reads per min) due to greater parallelisation of analysis jobs. For local analysis, four concurrent analysis jobs were run, whereas on the HPC six jobs were running in parallel. Despite the similar configuration, this local run was analysing reads ~87% faster than the average rate from the varying chunk size experiment described previously. This is partly due to the much shorter read length N50 of the *Clinical F3 M* run (5.67 kb) vs the Zymo gut mock run used for the chunk size experiment (19.57 kb). Analysis in HPC configuration lagged slightly behind the rate at which the GridION was producing basecalled pass reads (1,556 pass reads per min, Figure 4.4). However, the HPC configuration is easily scalable and matching the rate of production could be achieved by increasing the number of parallel jobs and total compute resources.

Despite the use of different basecalling models for the two runs, High Accuracy (HAC) on the GridION and Fast model on the MinION, the proportions of assigned reads at the species level produced by MARTi were extremely similar and showed very strong correlation (Pearson's r = 0.99, Figure 4.7). As the Fast model produces less accurate reads, the default read quality score filter in MinKNOW is slightly lower, 8 rather than 9. Regardless of Qscore filter applied by MinKNOW, MARTi allows users to configure their own pre-filter based on Qscore and read length. Both of the F3 sequencing runs used MARTi's default filter, minimum Qscore of 9 and minimum read length of 500 bp.

Real-time analysis with MARTi was also demonstrated on barcoded pool of 12 clinical faecal samples. The pool was sequenced on a GridION with a single MinION flow cell. Reads were basecalled and demultiplexed in real time and then analysed by the MARTi Engine running on an HPC. The taxonomic compositions of each of the 12 samples were successfully explored and compared during the sequencing run (Figure 4.6). The Samples card on the Compare page allowed samples to easily be ordered by disease classification before SVGs were exported from the stacked bar card. Currently, the MARTi GUI uses a palette of 11 colours for taxonomic plots. When using the GUI, reciprocal highlighting and tooltips make the plots easier to interpret. However, clarity of exported plots could be increased by adopting a larger colour palette.

4.4.4 Real-time AMR analysis of multiplexed clinical samples

MARTi's AMR walkout analysis identified a particular pathobiont, *Enterococcus faecium*, as a major contributor of AROs for many of the samples in the clinical faecal pool (Figure 4.8). *E. faecium* is a clinically important pathogen associated with opportunistic infection and is a known predictor of poor short-term survival in patients with decompensated cirrhosis and ACLF (Solé et al. 2021). The presence of this species resulted in a greater abundance of resistance ontologies focused on tetracycline, aminoglycoside, macrolide and diaminopyrimidine resistance. These results are similar to the resistance patterns described by Shamsaddini et al. (2021) for cirrhosis patients.

Only two of the clinical faecal samples did not have *E. faecium* in their top 20 walkout taxa after the first 100k reads were analysed, F18, a sample from the only stable cirrhotic patient in the pool, and F137, a sample taken from a patient with decompensated cirrhosis. The F137 sample had the fewest hits to AROs, with only 223 of the first 100k reads being assigned to an ARO. However, when analysis is expanded to all ~464k of the F137 reads that passed MARTi's filter, *E. faecium* was found to be amongst the top 10 walkout taxa. MARTi allows users to see when certain taxa or AMR genes were first detectable by dragging the chunk slider on either of the AMR cards. *E. faecium* only entered the top 20 walkout taxa for F137 after 148k reads were analysed, of which only 0.22% had hits to AROs. After analysing all of the reads (~444k) from the stable cirrhotic patient sample, F18, only 9 of the 2,386 ARO hits were assigned to *E. faecium* through walkout analysis, making it the only sample where *E. faecium* wasn't in the top 20 walkout taxa after all of the reads were analysed.

The AMR walkout donuts in Figure 4.8 were individually exported as SVGs from MARTi's Dashboard page for each sample in the pool and then assembled as one figure in Adobe Illustrator. This manual process highlighted the need for plots and export options for ARO hit and walkout results to be developed for the MARTi GUI's Compare page.

5 Chapter 5 – Discussion

Microorganisms have colonised almost every natural environment on Earth, and they play vital roles in both the biosphere and human health. However, only a small fraction of the microbial world, <1%, can be characterised with standard culture-based methods (Amann et al. 1995). High-throughput sequencing and associated computational methods have led to the development of shotgun metagenomics, which has revealed previously hidden diversity and function of complex communities. Nevertheless, accurate metagenomic classification and assembly remain challenging due to the limited information contained in individual sequences from the predominant short-read NGS technologies.

Long reads from third-generation sequencing technologies, such as ONT's nanopore-based platforms, have the potential to overcome many of the known problems associated with short-read metagenomics. Furthermore, ONT's technology offers the capability for both in-field and real-time metagenomics. Combined, the features of this new technology could lead to a better understanding of the composition and functional ecology of different microbial communities, facilitate the discovery of new species, enzymes, and biomolecules with potential applications in industry and medicine, and enable real-time analysis in time-critical situations, such as rapid infection diagnosis.

Yet, as the field is still in its infancy there is a lack of tools and methods available to make full use of the advantageous features of nanopore sequencing for metagenomics. In this project, two key features of this new technology (long reads and real-time data streaming) were drawn on to develop new analysis methods and tools for nanopore sequencing based metagenomics.

5.1 Reverse metagenomics

Chapter 2 described the development and testing of the RevMet (Reverse Metagenomics) method, which utilises long nanopore reads for semi-quantitative characterisation of samples containing a mixture of eukaryote species, which in this instance was mixed pollen species, without the need for complete reference genomes. The method was developed to address a crucial technical challenge for understanding plant-pollinator interactions, the identification and quantification of pollen species consumed by pollinators. Traditionally, this has been carried out using light microscopy to distinguish pollen based on grain morphology, a labour-intensive technique that requires expert knowledge. However, prior to this study, metabarcoding was the leading candidate for a low-cost, high-throughput, pollen characterisation method, but it has been shown to suffer from a lack of discriminatory power and is not fully quantitative due to PCR bias. In our pilot study, which as far as we know represents the first time nanopore sequencing was applied to bee-collected pollen, we demonstrated that RevMet can identify plant species present in mixed-species samples at proportions of DNA ≥1%, with both low false positives and false negatives, and was reliably 'semi-quantitative', that is, able to differentiate low- and high-frequency plant species, based on their DNA mass.

One of the known causes of quantitative errors in the study was variation in genome skim depths that were our flowering plant species database. Although 0.5x per reference was targeted, coverage varied across species, resulting in different powers of discrimination. Fortunately, we found that even very low-depth skims of 0.05x are useful for species detection and are probably still useful for differentiating rare from abundant species. As sequencing costs fall further, using higher coverage genome skims would be advisable for a more robust protocol. In cases where coverage for the reference skims varies greatly, they should be normalised to a uniform level.

RevMet is currently being developed further and applied at a much greater scale to study plant-pollinator interactions on commercial fruit farms (Kent et al. unpublished, see also engagement video on same project at https://royalsociety.org/science-events-and-lectures/2021/07/bees-favourite-flower/). Beyond mixed pollen samples, RevMet could have many other applications involving eukaryote species mixes, including the following:

- Herbivore diet analysis – RevMet could be used to characterise the diets of herbivores by applying the method to DNA extracted from their faeces. As a result, we could develop a better understanding of plant-herbivore interactions, which is essential for producing accurate trophic webs.
- Plant-fungal interactions - Arbuscular mycorrhizal fungi (AMF) are plant root symbionts associated with the roots of over 90% of plant species. Traditionally, studies on mycorrhizal diversity have relied on morphology and other phenotypic characteristics, which is time-consuming, expensive, and can be inaccurate due to similarities between AMF species. RevMet could be used as a cost-effective method to better understand plant-AMF interactions at a lower taxonomic level. This knowledge could lead to the ability to harness the symbiotic association for enhanced crop growth and yield.
- Airborne pollen monitoring – Characterising the presence and abundance of allergenic pollen in the air provides important information for hay fever suffers. As with bee-collected pollen identification, this task was traditionally carried out using light microscopy. Potentially, RevMet could be used as an alternative to morphological pollen identification to characterise pollen species filtered from the air.

Due to the lack of eukaryote reference data sets available at present, adopters of RevMet will likely need to create their own genome skims. With Illumina’s highest throughput sequencer, the NovaSeq 6000, the average 1x coverage 3Gb genome skim, almost twice the depth used in the RevMet study, would cost ~£50 each (250 bp PE reads from the SP flow cell). The genome sizes of plants used in this study ranged from to ~290 Mb to ~14.9 Gb. The per skim cost will be considerably lower for projects focussing on eukaryotes with smaller genomes such as Fungi, which have an average genome size of ~44 Mb (Ramos et al. 2015). Furthermore, the Earth BioGenome Project (EBP) aims to sequence all known eukaryote species before the end of 2028, and therefore more genomes should become freely available soon e.g., 2000 new UK species by end of 2022 from the Darwin Tree of Life project (part of EBP).

5.2 Development of MARTi

Currently, the only user-friendly tool capable of real-time classification is the EPI2ME service provided by ONT, which is not open and has significant limitations. Recognising the need for an open, extensible platform, we developed MARTi (**M**etagenomic **A**nalysis in **R**eal-**Ti**me), an open-source software tool that enables real-time analysis and visualisation of metagenomic sequencing data. MARTi provides a rapid, lightweight web interface that allows users to understand community composition and identify antimicrobial resistance (AMR) genes in real time. Chapter 3 provided an overview of the MARTi tool and its implementation, including descriptions of key analysis and visualisation algorithms.

MARTi consists of two main parts, the Engine and the GUI, and can be configured in multiple ways to suit the needs of the user. In HPC configuration, the MARTi Engine runs on a HPC server, whilst the MARTi GUI resides elsewhere. Analysis processes can be parallelised to a greater extent across a HPC making it less likely for analysis to fall behind sequencing, enabling larger databases to be used, and allowing multiple runs to be analysed simultaneously. In the local configuration, both the back and front end are run on the same device, typically a desktop or laptop, without the requirement of an internet connection. Combined with a minimalist in-field laboratory setup, MARTi enables real-time analysis at the site of sample collection. This was recently demonstrated by the Leggett group on Cromer pier, where all steps of the experiment, known as “Pier-seq”, including sampling, library preparation, sequencing and real-time analysis with MARTi, were carried out in-situ on the pier using battery power (Figure 5.1).

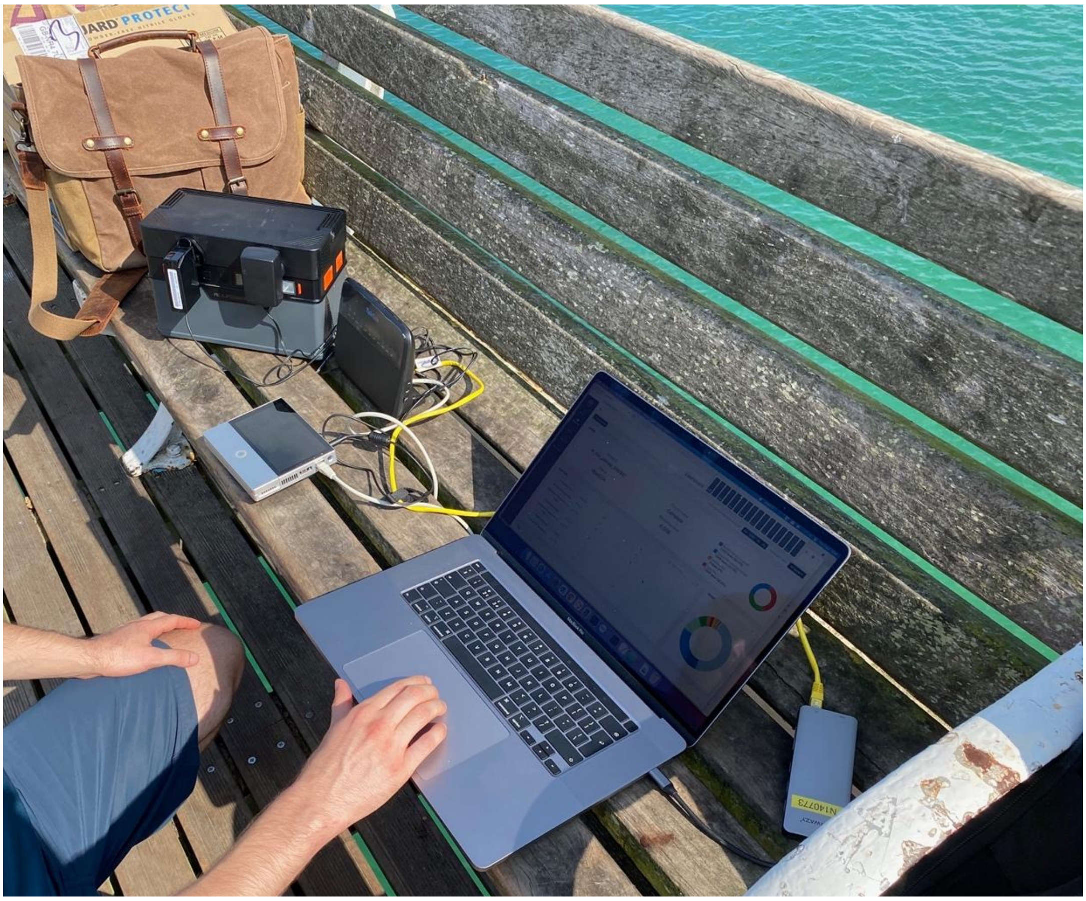


Figure 5.1 Combined with a minimalist in-field laboratory setup, MARTi enables real-time analysis in-situ. For the Pier-seq experiment, pictured here, all steps of the experiment from sampling to real-time analysis with MARTi, were carried out on Cromer pier. The MinION Mk1C produced chunks of basecalled reads that were synced to a laptop running MARTi in local configuration, where both the Engine and GUI were running on the same device.

The MARTi GUI features two main analysis modes: Dashboard, for analysing a single sample; and Compare, which allows multiple samples to be viewed side by side. Currently, the majority of plots available on the GUI are for exploring the taxonomic composition of samples, with the AMR table and AMR walkout analysis donut plot being the only forms of functional analysis. One area of future development for MARTi will be to expand its functional analysis offering, mapping reads to databases of gene groups such as KEGG to identify genes with known and annotated functions, and adding interactive functional analysis plots to explore the results. Other areas for future development include:

1. Improving existing plots by adding more options and features. For instance, a predictive curve could be added to the taxa accumulation plot to give the user an idea of how much more sequencing needs to be carried out in order to capture the species diversity within the sample. A larger colour palette could be introduced to increase the clarity of exported plots, especially the Compare page taxonomic plots.
2. Adding new comparison plots, for example, a comparison tree plot. This would allow multiple samples to be represented on a single tree for side-by-side comparisons on each node. A comparison heat map could also be developed, showing taxa on the y-axis and samples on the x-axis.
3. Support for alternative classification methods. MARTi classifies reads with a combination of BLAST and its own Lowest Common Ancestor (LCA) algorithm, producing similar classification results to other tools based on a BLAST-LCA approach such as MEGAN. The MARTi engine could be updated to support Centrifuge as an alternative to the current BLAST-LCA pipeline, allowing taxonomic analysis to be carried out more quickly and on devices with less memory.
4. Richness and evenness indices. MARTi could calculate and display metrics such as the Shannon–Wiener index and Simpson's index.
5. Clustering and correlation of samples. Support could be added for calculating and visualising multivariate analyses, such as principal component analysis and principal coordinates analysis, and correlations, such as Pearson's correlation coefficient.

With sufficient development effort, the ultimate vision is to provide a "plugin" interface to both the back-end and the front-end, which would enable third parties to simply develop new analysis pipelines and visualisations, accessible through the MARTi interface.

5.3 Demonstration of MARTi

Chapter 4 presented the results from testing MARTi during real experiments. Firstly, using a mock gut community of known composition, secondly, using clinical faecal samples from patients with advanced liver cirrhosis, and finally, faecal samples from healthy controls. A total of 98.7 gigabases of sequencing data was generated from five nanopore flow cells, giving an average yield of 19.7 Gb per flow cell. This is remarkable considering that typical MinION flow cell yields were in the region of 100s of megabases just six years ago (Ashton et al. 2015; Laver et al. 2015). The substantial increases in throughput and accuracy are a result of improvements ONT has made to their sequencing chemistry and analysis software.

MARTi was successfully used to analyse the taxonomic composition and AMR gene content of clinical microbiome samples in real time (Figure 4.3). Three live sequencing experiments were run to test different configurations of MARTi:

1. *Clinical pool* - a pool of 12 faecal samples from patients with cirrhosis, one stable cirrhotic, five decompensated cirrhotic and six acute-on-chronic liver failure (ACLF). The barcoded library was sequenced on a GridION with live High Accuracy (HAC) basecalling and demultiplexing. MARTi was run in HPC configuration.
2. *Clinical F3 M* - a single clinical faecal sample (F3), which originated from a patient with decompensated cirrhosis, was sequenced on the MinION Mk1C with live Fast model basecalling. MARTi was run in local configuration, with the Engine and GUI running on a single laptop.
3. *Clinical F3 G* run - the same F3 library from *Clinical F3 M* was also sequenced on the GridION with live HAC basecalling. Real-time analysis was carried out by MARTi in HPC configuration.

MARTi revealed that many of the patients exhibited gut microbiome dysbiosis with high proportions of *Enterococcaceae*, up to 96.8% for F40 (Figure 4.6). The *Enterococcus* genus is associated with complications in patients with end-stage liver disease (Llorente et al. 2017). The pathobiont *Enterococcus faecium* was identified as a major contributor of Antibiotic Resistance Ontologies in all of the clinical faecal samples except one (F18), the only stable cirrhotic patient sample (Figure 4.8). The presence of this species resulted in greater abundance of resistance ontologies focused on tetracycline, aminoglycoside, macrolide and diaminopyrimidine resistance. These results are similar to the resistance patterns observed by Shamsaddini et al. (2021) in patients with cirrhosis.

MARTi's walkout analysis assigned 98.5% of the ARO hits to taxonomies, with 41.1% of those being assigned at the species or strain level. Currently, MARTi relies on the uniqueness of the AMR gene sequence and flanking regions for low-level taxonomic assignments. If a read is not long enough to contain flanking regions, it is more likely have hits to multiple species and therefore be assigned to a higher taxonomic level by the LCA algorithm. Similarly, AMR genes based on plasmids can pose a challenge as the flanking regions can often have ambiguous taxonomic hits. However, as nanopore sequencing can capture base modifications, it might be possible in the future to use these modification patterns to assign a higher proportion of the AMR genes to species or strain level, including those on plasmids. This approach would require a database of AMR gene sequences with base modification data.

The *Clinical F3 G* run produced 2.24 million basecalled pass reads in the first 24 hours of GridION sequencing, an average rate of 1,556 pass reads per minute (Figure 4.4). Analysing in real-time on an HPC, MARTi BLAST searched batches of filtered reads in chunks of 4,000 to prokaryotic RefSeq genomes for taxonomic classification and also to CARD for AMR identification. Up to six chunks of reads were analysed in parallel. Within 24 hours, MARTi had analysed just over 1.83 million reads at an average rate of 1,273 reads per minute. Although the analysis rate lagged slightly behind the rate at which the GridION was producing basecalled pass reads, the HPC configuration is easily scalable and matching the rate of production could be achieved by increasing the number of parallel jobs and total compute resources.

Running in local configuration, MARTi analysed the *Clinical F3 M* run at ~740 reads per min with four concurrent jobs running on a 2021 2.3Ghz i9 8-core MacBook Pro. Although the rate of analysis was roughly half the rate of basecalled pass read production by the MinION Mk1C, analysis was fast enough for rapid taxonomic profiling with frequent GUI updates. Adding support for a memory efficient alignment-free classification tool such as Centrifuge to the MARTi engine as an alternative to the current BLAST-LCA pipeline would allow taxonomic analysis to be carried out more quickly and on lower-specification devices.

Increases in DNA sequencing throughput coupled with exponential growth of available reference genomes has already led to massive increases in the number of comparisons that need to be performed during analysis. The increase in demand on computational resources is very likely to outpace computer hardware development (Muir et al. 2016). Therefore, in order for real-time metagenomics to remain viable, particularly on PCs, analysis must become more efficient. One way of improving the analysis rate of MARTi on PC would be to adopt GPU-based analysis. Zhao and Chu (2014) demonstrated that GPU-accelerated (Nvidia GTX780) blastn and megablast algorithms run 1.56x and 7.15x faster respectively than multi-threaded NCBI BLAST running on 4 CPU cores (Intel Core i7-3820).

For in-field analysis, it might be possible to optimise MARTi to run on a compact, low-cost, high-performance, system such as the Jetson Xavier NX, which has 384 NVIDIA CUDA GPU Cores. It might also be possible to utilise some of the basecalling compute resources of a MinION Mk1C, which has 256 GPU cores, for MARTi analysis. This would enable sequencing, basecalling, and real-time metagenomic analysis on a single portable device.

Further acceleration could be achieved with FPGA-based (Field-Programmable Gate Array) analysis. For example, using a single TimeLogic J-series FPGA card, Tera-BLAST (an FPGA-accelerated implementation of the BLAST) runs up to 283.6x faster than NCBI BLAST+ running on a single core and up to 26.9x faster than NCBI BLAST+ running on 32 processor cores (TimeLogic 2013).

A pool of 12 faecal samples from healthy individuals was sequenced on a GridION and then analysed post-run by the MARTi Engine. The GUI was used to compare the clinical samples with the healthy samples. The species accumulation plot showed a pattern in species richness, with all of the healthy samples having a greater number of species detected within the first 100k reads than the cirrhotic clinical samples (Figure 4.9). An unequal variances t-test confirmed a significant difference in species richness between the healthy and clinical pools. This result is consistent with the observations from Solé et al. (2021), that cirrhosis is associated with a significant reduction in gene and metagenomic species richness and correlated with disease stages.

5.4 Future of nanopore sequencing

Nanopore sequencing technology is developing at an impressive rate and has seen massive increases in both accuracy and yield since first becoming publicly available in 2014. ONT continue to make improvements to the hardware, software, and chemistry of existing products, and are also developing new ones. New ASICs (application-specific integrated circuit) are in development to support new product lines such as Plongle and SmidgION.

The Plongle, “Plate dongle”, is a 96-well plate compatible sequencing device, with 96 individual, disposable flow cells. The device will enable users to carry out larger numbers of small sequencing runs in parallel, achieving a low cost-per-sample without the need for multiplexing. The device is designed to interface with liquid handling robots and multichannel pipettes for high-throughput, and even automated, preparation and loading of samples.

Unlike the bulky optical sensing approach adopted by PacBio, base detection based on ionic current measurements allows ONT to build extremely compact DNA sequencing platforms. The MinION is currently the only portable DNA sequencing platform available. However, ONT are developing an even smaller device, the SmidgION, designed for use with smartphones or other low power devices. This device could enable a broad range of in-field analyses including the following: real-time species ID, for authentication of food, drink, and timber; on-site analysis of environmental samples, such as water and soil; rapid clinical diagnosis of infectious disease; and pathogen monitoring during outbreaks.

It is likely that the SmidgION will be coupled with ONT’s cloud-based EPI2ME service for analysis. However, there is potential to develop an open-source MARTi mobile app that enables users to perform real-time, offline, analysis using the mobile device hardware. Samarakoon et al. (2020) recently developed the first ever smartphone application, Genopo, for nanopore sequencing analysis. The Android application was demonstrated on overlapping amplicon sequences, generating a complete consensus genome for SARS-CoV-2 in under 30 minutes on a range of popular smartphones.

In order to realise the full potential of portable sequencing devices, in-field library preparation methods also need to be developed. ONT have created a portable multi-purpose device, the VolTRAX V2, for automating laboratory processes upstream of nanopore sequencing, reducing hands-on time and enabling consistent library quality even in non-laboratory environments. The device includes heating elements for incubations, magnetic elements for bead-based clean-ups and a fluorescence detector for DNA quantitation. Currently, the small, USB-powered, device is restricted to PCR-free transposase-based library preparation protocols and does not carry out quantification or QC. In the future, the device will be capable of PCR, sample quantification, and running custom, user-programmed, protocols.

ONT currently relies on protein nanopores, but future generations of nanopore sequencing devices are likely to use more robust pores fabricated from synthetic materials (i.e., solid-state pores). Currently, ONT's flow cells are stable at ambient temperatures for 30 days, or up to 12 weeks if refrigerated. During sequencing pores can become damaged or even dislodged with aggressive unblocking. However, the superior mechanical and chemical stability of solid-state pores will enable flow cells to last much longer. These flow cells will also be more tolerant to washing and reuse.

At present, each nanopore has its own electrode connected to a channel in the sensor array chip and ionic current measurements are collected ten thousand times per second. This is more than sufficient for the current rate of DNA translocation across the pore (~420 bp/s), but proposed alternatives, such as graphene-based nano-gap or edge state detectors, could enable detection at much greater rates.

A combination of solid-state pores and improved sensing could enable sequencing of unmodified nucleotide molecules (i.e., no added sequencing adaptors and motor protein) at a rate of hundreds of thousands to millions of base pairs per second. This library-prep free, solid-state, sequencing technology would enable the development of sequencing sensors that could be coupled with real-time analysis for continuous environmental monitoring.

6 Bibliography

The jQuery Team (2018). jQuery (v3.3.1). https://github.com/jquery/jquery/releases/tag/3.3.1

OpenJS Foundation (2020). Node.js (v12.14.1). https://github.com/nodejs/node/releases/tag/v12.14.1

Alcock, B. P., Raphenya, A. R., Lau, T. T. Y., Tsang, K. K., Bouchard, M., Edalatmand, A., Huynh, W., Nguyen, A. L. V., Cheng, A. A., Liu, S. H., Min, S. Y., Miroshnichenko, A., Tran, H. K., Werfalli, R. E., Nasir, J. A., Oloni, M., Speicher, D. J., Florescu, A., Singh, B., Faltyn, M., Hernandez-Koutoucheva, A., Sharma, A. N., Bordeleau, E., Pawlowski, A. C., Zubyk, H. L., Dooley, D., Griffiths, E., Maguire, F., Winsor, G. L., Beiko, R. G., Brinkman, F. S. L., Hsiao, W. W. L., Domselaar, G. V. and McArthur, A. G. (2020). CARD 2020: antibiotic resistome surveillance with the comprehensive antibiotic resistance database. *Nucleic Acids Research*, 48(D1), D517-D525.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W. and Lipman, D. J. (1990). Basic local alignment search tool. *Journal of molecular biology*, 215(3), 403-410.

Amann, R. I., Ludwig, W. and Schleifer, K.-H. (1995). Phylogenetic identification and in situ detection of individual microbial cells without cultivation. *Microbiological reviews*, 59(1), 143-169.

Amarasinghe, S. L., Su, S., Dong, X. Y., Zappia, L., Ritchie, M. E. and Gouil, Q. (2020). Opportunities and challenges in long-read sequencing data analysis. *Genome Biology*, 21(1).

Arroyo, V., Moreau, R., Kamath, P. S., Jalan, R., Gines, P., Nevens, F., Fernandez, J., To, U., Garcia-Tsao, G. and Schnabl, B. (2016). Acute-on-chronic liver failure in cirrhosis. *Nat Rev Dis Primers*, 2, 16041.

Ashton, P. M., Nair, S., Dallman, T., Rubino, S., Rabsch, W., Mwaigwisya, S., Wain, J. and O'Grady, J. (2015). MinION nanopore sequencing identifies the position and structure of a bacterial antibiotic resistance island. *Nature Biotechnology*, 33(3), 296-+.

Ayling, M., Clark, M. D. and Leggett, R. M. (2020). New approaches for metagenome assembly with short reads. *Briefings in Bioinformatics*, 21(2), 584-594.

Bell, K. L., Fowler, J., Burgess, K. S., Dobbs, E. K., Gruenewald, D., Lawley, B., Morozumi, C. and Brosi, B. J. (2017). Applying Pollen DNA Metabarcoding to the Study of Plant-Pollinator Interactions. *Applications in Plant Sciences*, 5(6).

Bhattacharyya, S., Dawson, D. A., Hipperson, H. and Ishtiaq, F. (2019). A diet rich in C3 plants reveals the sensitivity of an alpine mammal to climate change. *Mol Ecol*, 28(2), 250-265.

Bommarco, R., Kleijn, D. and Potts, S. G. (2013). Ecological intensification: harnessing ecosystem services for food security. *Trends Ecol Evol*, 28(4), 230-238.

Bostock, M.(2016). D3.js (v3.5.17). https://github.com/d3/d3/releases/tag/v3.5.17

Breitwieser, F. P. and Salzberg, S. L. (2020). Pavian: interactive analysis of metagenomics data for microbiome studies and pathogen identification. *Bioinformatics*, 36(4), 1303-1304.

Buchfink, B., Xie, C. and Huson, D. H. (2015). Fast and sensitive protein alignment using DIAMOND. *Nature Methods*, 12(1), 59-60.

Burkle, L. A., Marlin, J. C. and Knight, T. M. (2013). Plant-pollinator interactions over 120 years: loss of species, co-occurrence, and function. *Science*, 339(6127), 1611-1615.

Carvalheiro, L. G., Biesmeijer, J. C., Benadi, G., Frund, J., Stang, M., Bartomeus, I., Kaiser-Bunbury, C. N., Baude, M., Gomes, S. I. F., Merckx, V., Baldock, K. C. R., Bennett, A. T. D., Boada, R., Bommarco, R., Cartar, R., Chacoff, N., Danhardt, J., Dicks, L. V., Dormann, C. F., Ekroos, J., Henson, K. S. E., Holzschuh, A., Junker, R. R., Lopezaraiza-Mikel, M., Memmott, J., Montero-Castano, A., Nelson, I. L., Petanidou, T., Power, E. F., Rundlof, M., Smith, H. G., Stout, J. C., Temitope, K., Tscharntke, T., Tscheulin, T., Vila, M. and Kunin, W. E. (2014). The potential for indirect effects between co-flowering plants via shared pollinators depends on resource abundance, accessibility and relatedness. *Ecology Letters*, 17(11), 1389-1399.

Castro-Wallace, S. L., Chiu, C. Y., John, K. K., Stahl, S. E., Rubins, K. H., McIntyre, A. B. R., Dworkin, J. P., Lupisella, M. L., Smith, D. J., Botkin, D. J., Stephenson, T. A., Juul, S., Turner, D. J., Izquierdo, F., Federman, S., Stryke, D., Somasekar, S., Alexander, N., Yu, G. X., Mason, C. E. and Burton, A. S. (2017). Nanopore DNA Sequencing and Genome Assembly on the International Space Station. *Scientific Reports*, 7.

Charalampous, T., Kay, G. L., Richardson, H., Aydin, A., Baldan, R., Jeanes, C., Rae, D., Grundy, S., Turner, D. J., Wain, J., Leggett, R. M., Livermore, D. M. and O'Grady, J. (2019). Nanopore metagenomics enables rapid clinical diagnosis of bacterial lower respiratory infection. *Nature Biotechnology*, 37(7), 783-792.

Chen, Z., Erickson, D. L. and Meng, J. (2021). Polishing the Oxford Nanopore long-read assemblies of bacterial pathogens with Illumina short reads to improve genomic analyses. *Genomics*, 113(3), 1366-1377.

Chiu, C. Y. and Miller, S. A. (2019). Clinical metagenomics. *Nature Reviews Genetics*, 20(6), 341-355.

Cusco, A., Perez, D., Vines, J., Fabregas, N. and Francino, O. (2021). Long-read metagenomics retrieves complete single-contig bacterial genomes from canine feces. *Bmc Genomics*, 22(1).

De, R. (2019). Metagenomics: aid to combat antimicrobial resistance in diarrhea. *Gut Pathogens*, 11(1).

Dicks, L. V., Abrahams, A., Atkinson, J., Biesmeijer, J., Bourn, N., Brown, C., Brown, M. J. F., Carvell, C., Connolly, C., Cresswell, J. E., Croft, P., Darvill, B., De Zylva, P., Effingham, P., Fountain, M., Goggin, A., Harding, D., Harding, T., Hartfield, C., Heard, M. S., Heathcote, R., Heaver, D., Holland, J., Howe, M., Hughes, B., Huxley, T., Kunin, W. E., Little, J., Mason, C., Memmott, J., Osborne, J., Pankhurst, T., Paxton, R. J., Pocock, M. J. O., Potts, S. G., Power, E. F., Raine, N. E., Ranelagh, E., Roberts, S., Saunders, R., Smith, K., Smith, R. M., Sutton, P., Tilley, L. A. N., Tinsley, A., Tonhasca, A., Vanbergen, A. J., Webster, S., Wilson, A. and Sutherland, W. J. (2013). Identifying key knowledge needs for evidence-based conservation of wild insect pollinators: a collaborative cross-sectoral exercise. *Insect Conservation and Diversity*, 6(3), 435-446.

Dicks, L. V., Baude, M., Roberts, S. P. M., Phillips, J., Green, M. and Carvell, C. (2015). How much flower-rich habitat is enough for wild pollinators? Answering a key policy question with incomplete knowledge. *Ecological Entomology*, 40, 22-35.

Dormann, C. F., Fründ, J., Blüthgen, N. and Gruber, B. (2009). Indices, graphs and null models: analyzing bipartite ecological networks. *The Open Ecology Journal*, 2(1).

Driscoll, C. B., Otten, T. G., Brown, N. M. and Dreher, T. W. (2017). Towards long-read metagenomics: complete assembly of three novel genomes from bacteria dependent on a diazotrophic cyanobacterium in a freshwater lake co-culture. *Standards in Genomic Sciences*, 12.

Escobar-Zepeda, A., de Leon, A. V. P. and Sanchez-Flores, A. (2015). The Road to Metagenomics: From Microbiology to DNA Sequencing Technologies and Bioinformatics. *Frontiers in Genetics*, 6.

Ferrer, M., Martinez-Martinez, M., Bargiela, R., Streit, W. R., Golyshina, O. V. and Golyshin, P. N. (2016). Estimating the success of enzyme bioprospecting through metagenomics: current status and future trends. *Microbial Biotechnology*, 9(1), 22-34.

Forup, M. L. and Memmott, J. (2005). The restoration of plant-pollinator interactions in hay meadows. *Restoration Ecology*, 13(2), 265-274.

Gilbert, J. A. and Dupont, C. L. (2011). Microbial Metagenomics: Beyond the Genome. *Annual Review of Marine Science, Vol 3*, 3, 347-371.

Grossart, H. P., Massana, R., McMahon, K. D. and Walsh, D. A. (2020). Linking metagenomics to aquatic microbial ecology and biogeochemical cycles. *Limnology and Oceanography*, 65, S2-S20.

Grüter, C. and Ratnieks, F. L. (2011). Flower constancy in insect pollinators: Adaptive foraging behaviour or cognitive limitation? *Communicative & Integrative Biology*, 4(6), 633-636.

Hollingsworth, P. M., Li, D. Z., van der Bank, M. and Twyford, A. D. (2016). Telling plant species apart with DNA: from barcodes to genomes. *Philosophical Transactions of the Royal Society B-Biological Sciences*, 371(1702).

Huson, D. H., Auch, A. F., Qi, J. and Schuster, S. C. (2007). MEGAN analysis of metagenomic data. *Genome Research*, 17(3), 377-386.

Ji, Y. Q., Ashton, L., Pedley, S. M., Edwards, D. P., Tang, Y., Nakamura, A., Kitching, R., Dolman, P. M., Woodcock, P., Edwards, F. A., Larsen, T. H., Hsu, W. W., Benedick, S., Hamer, K. C., Wilcove, D. S., Bruce, C., Wang, X. Y., Levi, T., Lott, M., Emerson, B. C. and Yu, D. W. (2013). Reliable, verifiable and efficient monitoring of biodiversity via metabarcoding. *Ecology Letters*, 16(10), 1245-1257.

Ji, Y. Q., Huotari, T., Roslin, T., Schmidt, N. M., Wang, J. X., Yu, D. W. and Ovaskainen, O. (2020). SPIKEPIPE: A metagenomic pipeline for the accurate quantification of eukaryotic species occurrences and intraspecific abundance change using DNA barcodes or mitogenomes. *Molecular Ecology Resources*, 20(1), 256-267.

Johnson, S. S., Zaikova, E., Goerlitz, D. S., Bai, Y. and Tighe, S. W. (2017). Real-Time DNA Sequencing in the Antarctic Dry Valleys Using the Oxford Nanopore Sequencer. *J Biomol Tech*, 28(1), 2-7.

Jones, M. B., Highlander, S. K., Anderson, E. L., Li, W. Z., Dayrit, M., Klitgord, N., Fabani, M. M., Seguritan, V., Green, J., Pride, D. T., Yooseph, S., Biggs, W., Nelson, K. E. and Venter, J. C. (2015). Library preparation methodology can influence genomic and functional predictions in human microbiome research. *Proceedings of the National Academy of Sciences of the United States of America*, 112(45), 14024-14029.

Juul, S., Izquierdo, F., Hurst, A., Dai, X., Wright, A., Kulesha, E., Pettett, R. and Turner, D. J. (2015). What's in my pot? Real-time species identification on the MinION™. *bioRxiv*, 030742.

Karsenti, E., Acinas, S. G., Bork, P., Bowler, C., De Vargas, C., Raes, J., Sullivan, M., Arendt, D., Benzoni, F., Claverie, J. M., Follows, M., Gorsky, G., Hingamp, P., Iudicone, D., Jaillon, O., Kandels-Lewis, S., Krzic, U., Not, F., Ogata, H., Pesant, S., Reynaud, E. G., Sardet, C., Sieracki, M. E., Speich, S., Velayoudon, D., Weissenbach, J., Wincker, P. and Consortium, T. O. (2011). A Holistic Approach to Marine Eco-Systems Biology. *Plos Biology*, 9(10).

Keller, A., Danner, N., Grimmer, G., Ankenbrand, M., von der Ohe, K., von der Ohe, W., Rost, S., Hartel, S. and Steffan-Dewenter, I. (2015). Evaluating multiplexed next-generation sequencing as a method in palynology for mixed pollen samples. *Plant Biology*, 17(2), 558-566.

Khansari, E., Zarre, S., Alizadeh, K., Attar, F., Aghabeigi, F. and Salmaki, Y. (2012). Pollen morphology of Campanula (Campanulaceae) and allied genera in Iran with special focus on its systematic implication. *Flora*, 207(3), 203-211.

Kieffer, R., Tavan, C., O'Neal, A. J., Voyer, V. and Shtylman, R.(2020). uuid (v8.3.2). https://github.com/uuidjs/uuid/releases/tag/v8.3.2

Kim, D., Song, L., Breitwieser, F. P. and Salzberg, S. L. (2016). Centrifuge: rapid and sensitive classification of metagenomic sequences. *Genome Research*, 26(12), 1721-1729.

Kinkar, L., Gasser, R. B., Webster, B. L., Rollinson, D., Littlewood, D. T. J., Chang, B. C., Stroehlein, A. J., Korhonen, P. K. and Young, N. D. (2021). Nanopore sequencing resolves elusive long tandem-repeat regions in mitochondrial genomes. *International journal of molecular sciences*, 22(4), 1811.

Klein, A. M., Vaissiere, B. E., Cane, J. H., Steffan-Dewenter, I., Cunningham, S. A., Kremen, C. and Tscharntke, T. (2007). Importance of pollinators in changing landscapes for world crops. *Proceedings of the Royal Society B-Biological Sciences*, 274(1608), 303-313.

Kolmogorov, M., Bickhart, D. M., Behsaz, B., Gurevich, A., Rayko, M., Shin, S. B., Kuhn, K., Yuan, J., Polevikov, E., Smith, T. P. L. and Pevzner, P. A. (2020). metaFlye: scalable long-read metagenome assembly using repeat graphs. *Nature Methods*, 17(11), 1103-1110.

Koren, S., Schatz, M. C., Walenz, B. P., Martin, J., Howard, J. T., Ganapathy, G., Wang, Z., Rasko, D. A., McCombie, W. R., Jarvis, E. D. and Phillippy, A. M. (2012). Hybrid error correction and de novo assembly of single-molecule sequencing reads. *Nature Biotechnology*, 30(7), 692-700.

Kraaijeveld, K., De Weger, L. A., Garcia, M. V., Buermans, H., Frank, J., Hiemstra, P. S. and Den Dunnen, J. T. (2015). Efficient and sensitive identification and quantification of airborne pollen using next-generation DNA sequencing. *Molecular Ecology Resources*, 15(1), 8-16.

Kress, W. J., Garcia-Robledo, C., Uriarte, M. and Erickson, D. L. (2015). DNA barcodes for ecology, evolution, and conservation. *Trends Ecol Evol*, 30(1), 25-35.

Lamb, P. D., Hunter, E., Pinnegar, J. K., Creer, S., Davies, R. G. and Taylor, M. I. (2019). How quantitative is metabarcoding: A meta-analytical approach. *Molecular Ecology*, 28(2), 420-430.

Laver, T., Harrison, J., O'Neill, P. A., Moore, K., Farbos, A., Paszkiewicz, K. and Studholme, D. J. (2015). Assessing the performance of the Oxford Nanopore Technologies MinION. *Biomol Detect Quantif*, 3, 1-8.

Leggett, R. M., Alcon-Giner, C., Heavens, D., Caim, S., Brook, T. C., Kujawska, M., Martin, S., Peel, N., Acford-Palmer, H., Hoyles, L., Clarke, P., Hall, L. J. and Clark, M. D. (2020). Rapid MinION profiling of preterm microbiota and antimicrobial-resistant pathogens. *Nature Microbiology*, 5(3), 430-442.

Leggett, R. M. and Clark, M. D. (2017). A world of opportunities with nanopore sequencing. *Journal of Experimental Botany*, 68(20), 5419-5429.

Leggett, R. M., Clavijo, B. J., Clissold, L., Clark, M. D. and Caccamo, M. (2014). NextClip: an analysis and read preparation tool for Nextera Long Mate Pair libraries. *Bioinformatics*, 30(4), 566-568.

Li, H. (2013). Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *arXiv preprint arXiv:1303.3997*.

Li, H. (2018). Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, 34(18), 3094-3100.

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G. and Durbin, R. (2009). The sequence alignment/map format and SAMtools. *Bioinformatics*, 25(16), 2078-2079.

Li, X. W., Yang, Y., Henry, R. J., Rossetto, M., Wang, Y. T. and Chen, S. L. (2015). Plant DNA barcoding: from gene to genome. *Biological Reviews*, 90(1), 157-166.

Llorente, C., Jepsen, P., Inamine, T., Wang, L., Bluemel, S., Wang, H. J., Loomba, R., Bajaj, J. S., Schubert, M. L., Sikaroodi, M., Gillevet, P. M., Xu, J., Kisseleva, T., Ho, S. B., DePew, J., Du, X., Sorensen, H. T., Vilstrup, H., Nelson, K. E., Brenner, D. A., Fouts, D. E. and Schnabl, B. (2017). Gastric acid suppression promotes alcoholic liver disease by inducing overgrowth of intestinal Enterococcus. *Nat Commun*, 8(1), 837.

Long, E. Y. and Krupke, C. H. (2016). Non-cultivated plants present a season-long route of pesticide exposure for honey bees. *Nature Communications*, 7.

Loose, M., Malla, S. and Stout, M. (2016). Real-time selective sequencing using nanopore technology. *Nature Methods*, 13(9), 751-754.

Lucas, A., Bodger, O., Brosi, B. J., Ford, C. R., Forman, D. W., Greig, C., Hegarty, M., Neyland, P. J. and de Vere, N. (2018). Generalisation and specialisation in hoverfly (Syrphidae) grassland pollen transport networks revealed by DNA metabarcoding. *Journal of Animal Ecology*, 87(4), 1008-1021.

Martin, M. (2011). Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet. journal*, 17(1), 10-12.

Martin, S., Heavens, D., Lan, Y., Horsfield, S., Clark, M. D. and Leggett, R. M. (2021). Nanopore adaptive sampling: a tool for enrichment of low abundance species in metagenomic samples. *bioRxiv*, 2021.2005.2007.443191.

Menegon, M., Cantaloni, C., Rodriguez-Prieto, A., Centomo, C., Abdelfattah, A., Rossato, M., Bernardi, M., Xumerle, L., Loader, S. and Delledonne, M. (2017). On site DNA barcoding by nanopore sequencing. *Plos One*, 12(10).

Miller, P.(2020). Chokidar (v3.4.2). https://github.com/paulmillr/chokidar/releases/tag/3.4.2

Minot, S. S., Krumm, N. and Greenfield, N. B. (2015). One codex: a sensitive and accurate data platform for genomic microbial identification. *bioRxiv*, 027607.

Morales, C. L. and Traveset, A. (2008). Interspecific pollen transfer: Magnitude, prevalence and consequences for plant fitness. *Critical Reviews in Plant Sciences*, 27(4), 221-238.

Morandin, L. A. and Kremen, C. (2013). Hedgerow restoration promotes pollinator populations and exports native bees to adjacent fields. *Ecological Applications*, 23(4), 829-839.

Moss, E. L., Maghini, D. G. and Bhatt, A. S. (2020). Complete, closed bacterial genomes from microbiomes using nanopore sequencing. *Nature Biotechnology*, 38(6), 701-707.

Muir, P., Li, S., Lou, S., Wang, D., Spakowicz, D. J., Salichos, L., Zhang, J., Weinstock, G. M., Isaacs, F., Rozowsky, J. and Gerstein, M. (2016). The real cost of sequencing: scaling computation to keep pace with data generation. *Genome Biology*, 17, 53.

Nayfach, S. and Pollard, K. S. (2016). Toward Accurate and Quantitative Comparative Metagenomics. *Cell*, 166(5), 1103-1116.

Ollerton, J., Winfree, R. and Tarrant, S. (2011). How many flowering plants are pollinated by animals? *Oikos*, 120(3), 321-326.

Otto, M. and Thornton, J.(2019). Bootstrap (v4.3.1). https://github.com/twbs/bootstrap/releases/tag/v4.3.1

Payne, A., Holmes, N., Clarke, T., Munro, R., Debebe, B. J. and Loose, M. (2021). Readfish enables targeted nanopore sequencing of gigabase-sized genomes. *Nature Biotechnology*, 39(4), 442-450.

Pearman, W. S., Freed, N. E. and Silander, O. K. (2020). Testing the advantages and disadvantages of short- and long- read eukaryotic metagenomics using simulated reads. *Bmc Bioinformatics*, 21(1).

Pellicer, J. and Leitch, I. J. (2020). The Plant DNA C-values database (release 7.1): an updated online repository of plant genome size data for comparative studies. *New Phytol*, 226(2), 301-305.

Perez-Sepulveda, B. M., Heavens, D., Pulford, C. V., Predeus, A. V., Low, R., Webster, H., Schudoma, C., Rowe, W., Lipscombe, J. and Watkins, C. (2020). An accessible, efficient and global approach for the large-scale sequencing of bacterial genomes. *bioRxiv*.

Pornon, A., Escaravage, N., Burrus, M., Holota, H., Khimoun, A., Mariette, J., Pellizzari, C., Iribar, A., Etienne, R., Taberlet, P., Vidal, M., Winterton, P., Zinger, L. and Andalo, C. (2016). Using metabarcoding to reveal and quantify plant-pollinator interactions. *Scientific Reports*, 6.

Potts, S. G., Biesmeijer, J. C., Kremen, C., Neumann, P., Schweiger, O. and Kunin, W. E. (2010). Global pollinator declines: trends, impacts and drivers. *Trends in Ecology & Evolution*, 25(6), 345-353.

Potts, S. G., Imperatriz-Fonseca, V., Ngo, H. T., Aizen, M. A., Biesmeijer, J. C., Breeze, T. D., Dicks, L. V., Garibaldi, L. A., Hill, R., Settele, J. and Vanbergen, A. J. (2016a). Safeguarding pollinators and their values to human well-being. *Nature*, 540(7632), 220-229.

Potts, S. G., Imperatriz-Fonseca, V., Ngo, H. T., Biesmeijer, J. C., Breeze, T. D., Dicks, L. V., Garibaldi, L. A., Hill, R., Settele, J. and Vanbergen, A. J. (2016b). The assessment report on pollinators, pollination and food production: summary for policymakers.

Quick, J., Nicholls, S. and Loman, N. (2019). The 'Three Peaks' faecal DNA extraction method for long-read sequencing. *protocols.io*.

Ramos, A. P., Tavares, S., Tavares, D., Silva Mdo, C., Loureiro, J. and Talhinhas, P. (2015). Flow cytometry reveals that the rust fungus, Uromyces bidentis (Pucciniales), possesses the largest fungal genome reported--2489 Mbp. *Mol Plant Pathol*, 16(9), 1006-1010.

Ratnasingham, S. and Hebert, P. D. N. (2007). BOLD: The Barcode of Life Data System (www.barcodinglife.org). *Molecular Ecology Notes*, 7(3), 355-364.

Rauch, G.(2018). Socket.IO (v2.1.1). https://github.com/socketio/socket.io/releases/tag/2.1.1

Richardson, J. P.(2020). fs-extra (v9.0.1). https://github.com/jprichardson/node-fs-extra/releases/tag/9.0.1

Richardson, R. T., Lin, C. H., Sponsler, D. B., Quijia, J. O., Goodell, K. and Johnson, R. M. (2015). Application of Its2 Metabarcoding to Determine the Provenance of Pollen Collected by Honey Bees in an Agroecosystem. *Applications in Plant Sciences*, 3(1).

Samarakoon, H., Punchihewa, S., Senanayake, A., Hammond, J. M., Stevanovski, I., Ferguson, J. M., Ragel, R., Gamaarachchi, H. and Deveson, I. W. (2020). Genopo: a nanopore sequencing analysis toolkit for portable Android devices. *Communications biology*, 3(1), 1-5.

Sanderson, N. D., Street, T. L., Foster, D., Swann, J., Atkins, B. L., Brent, A. J., McNally, M. A., Oakley, S., Taylor, A., Peto, T. E. A., Crook, D. W. and Eyre, D. W. (2018). Real-time analysis of nanopore-based metagenomic sequencing from infected orthopaedic devices. *Bmc Genomics*, 19(1), 714.

Schroter, K., Wemheuer, B., Pena, R., Schoning, I., Ehbrecht, M., Schall, P., Ammer, C., Daniel, R. and Polle, A. (2019). Assembly processes of trophic guilds in the root mycobiome of temperate forests. *Molecular Ecology*, 28(2), 348-364.

Sereika, M., Kirkegaard, R. H., Karst, S. M., Michaelsen, T. Y., Sorensen, E. A., Wollenberg, R. D. and Albertsen, M. (2022). Oxford Nanopore R10.4 long-read sequencing enables the generation of near-finished bacterial genomes from pure cultures and metagenomes without short-read or reference polishing. *Nature Methods*, 19(7), 823-826.

Shamsaddini, A., Gillevet, P. M., Acharya, C., Fagan, A., Gavis, E., Sikaroodi, M., McGeorge, S., Khoruts, A., Albhaisi, S., Fuchs, M., Sterling, R. K. and Bajaj, J. S. (2021). Impact of Antibiotic Resistance Genes in Gut Microbiome of Patients With Cirrhosis. *Gastroenterology*, 161(2), 508-521 e507.

Sharpton, T. J. (2014). An introduction to the analysis of shotgun metagenomic data. *Frontiers in Plant Science*, 5.

Shneiderman, B. (1992). Tree visualization with tree-maps: 2-d space-filling approach. *ACM Trans. Graph.*, 11(1), 92–99.

Sickel, W., Ankenbrand, M. J., Grimmer, G., Holzschuh, A., Hartel, S., Lanzen, J., Steffan-Dewenter, I. and Keller, A. (2015). Increased efficiency in identifying mixed pollen samples by meta-barcoding with a dual-indexing approach. *Bmc Ecology*, 15.

Solé, C., Guilly, S., Da Silva, K., Llopis, M., Le-Chatelier, E., Huelin, P., Carol, M., Moreira, R., Fabrellas, N. and De Prada, G. (2021). Alterations in gut microbiome in cirrhosis as assessed by quantitative metagenomics: relationship with acute-on-chronic liver failure and prognosis. *Gastroenterology*, 160(1), 206-218. e213.

Somerville, V., Lutz, S., Schmid, M., Frei, D., Moser, A., Irmler, S., Frey, J. E. and Ahrens, C. H. (2019). Long-read based de novo assembly of low-complexity metagenome samples results in finished genomes and reveals insights into strain diversity and an active phage system. *Bmc Microbiology*, 19.

SpryMedia(2018). DataTables (v1.10.19). https://github.com/DataTables/DataTables/releases/tag/1.10.19

Straub, S. C. K., Parks, M., Weitemier, K., Fishbein, M., Cronn, R. C. and Liston, A. (2012). Navigating the Tip of the Genomic Iceberg: Next-Generation Sequencing for Plant Systematics. *American Journal of Botany*, 99(2), 349-364.

Team, R. C.(2018). R: A language and environment for statistical computing http://www.R-project.org/

TimeLogic (2013). Accelerated BLAST Performance with Tera-BLAST. Technical Note.

Tittonell, P. (2014). Ecological intensification of agriculture - sustainable by nature. *Current Opinion in Environmental Sustainability*, 8, 53-61.

Tourancheau, A., Mead, E. A., Zhang, X. S. and Fang, G. (2021). Discovering multiple types of DNA methylation from bacteria and microbiome using nanopore sequencing. *Nature Methods*, 18(5), 491-498.

Urban, L., Holzer, A., Baronas, J. J., Hall, M. B., Braeuninger-Weimer, P., Scherm, M. J., Kunz, D. J., Perera, S. N., Martin-Herranz, D. E., Tipper, E. T., Salter, S. J. and Stammnitz, M. R. (2021). Freshwater monitoring by nanopore sequencing. *Elife*, 10.

Vanbergen, A. J., Baude, M., Biesmeijer, J. C., Britton, N. F., Brown, M. J. F., Brown, M., Bryden, J., Budge, G. E., Bull, J. C., Carvell, C., Challinor, A. J., Connolly, C. N., Evans, D. J., Feil, E. J., Garratt, M. P., Greco, M. K., Heard, M. S., Jansen, V. A. A., Keeling, M. J., Kunis, W. E., Marris, G. C., Memmott, J., Murray, J. T., Nicolson, S. W., Osborne, J. L., Paxton, R. J., Pirk, C. W. W., Polce, C., Potts, S. G., Priest, N. K., Raine, N. E., Roberts, S., Ryabov, E. V., Shafir, S., Shirley, M. D. F., Simpson, S. J., Stevenson, P. C., Stone, G. N., Termansen, M., Wright, G. A. and Initiative, I. P. (2013). Threats to an ecosystem service: pressures on pollinators. *Frontiers in Ecology and the Environment*, 11(5), 251-259.

Watson, M. and Warr, A. (2019). Errors in long-read assemblies can critically affect protein prediction. *Nature Biotechnology*, 37(2), 124-126.

Wilson, D. C.(2018). Express.js (v4.16.4). https://github.com/expressjs/express/releases/tag/4.16.4

Wilson, M. C. and Piel, J. (2013). Metagenomic Approaches for Exploiting Uncultivated Bacteria as a Resource for Novel Biosynthetic Enzymology. *Chemistry & Biology*, 20(5), 636-647.

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

Wood, T. J., Holland, J. M. and Goulson, D. (2015). Pollinator-friendly management does not increase the diversity of farmland bees and wasps. *Biological Conservation*, 187, 120-126.

Woodgate, J. L., Makinson, J. C., Lim, K. S., Reynolds, A. M. and Chittka, L. (2016). Life-Long Radar Tracking of Bumblebees. *Plos One*, 11(8).

Wu, X. W., Luo, H., Xu, F., Ge, C. T., Li, S. T., Deng, X. Y., Wiedmann, M., Baker, R. C., Stevenson, A., Zhang, G. T. and Tang, S. L. (2021). Evaluation of Salmonella Serotype Prediction With Multiplex Nanopore Sequencing. *Frontiers in Microbiology*, 12.

Zhao, K. and Chu, X. (2014). G-BLASTN: accelerating nucleotide alignment by graphics processors. *Bioinformatics*, 30(10), 1384-1391.

Zhou, X., Li, Y. Y., Liu, S. L., Yang, Q., Su, X., Zhou, L. L., Tang, M., Fu, R. B., Li, J. G. and Huang, Q. F. (2013). Ultra-deep sequencing enables high-fidelity recovery of biodiversity for bulk arthropod samples without PCR amplification. *Gigascience*, 2.

7 Appendices

Appendix 1 Estimated genome sizes and genome coverage for the plant reference skims

Plant species	Estimated genome size (Mbp)	Raw PE read counts	Post-processing PE read counts	Post-processing estimated coverage (x)
Achillea millefolium	7,482	9,501,046	6,580,562	0.44
Anagallis arvensis	1,712	2,085,605	1,934,932	0.57
Ballota nigra	2,425	2,636,800	2,242,563	0.46
Bromus commutatus	10,636	9,459,893	7,033,860	0.33
Bryonia dioica	1,614	1,873,028	1,741,086	0.54
Centaurea nigra	1,760	2,561,721	2,320,727	0.66
Chaerophyllum temulum	4,144	4,977,454	4,374,009	0.53
Cirsium arvense	1,389	1,778,262	1,636,918	0.59
Conium maculatum	4,144	4,622,584	4,117,236	0.5
Convolvulus arvensis	1,736	2,313,746	2,161,875	0.62
Crepis capillaris	2,054	2,421,660	2,192,821	0.53
Digitalis purpurea	1,198	1,386,853	1,293,554	0.54
Elymus caninus	8,362	8,149,150	5,874,468	0.35
Epilobium hirsutum	293	526,202	507,820	0.87
Galium verum	1,845	2,662,086	2,358,137	0.64
Geranium dissectum	1,283	1,509,064	1,427,315	0.56
Geranium robertianum	1,283	1,381,657	1,301,665	0.51
Holcus lanatus	1,663	1,962,801	1,794,155	0.54
Hypericum perforatum	766	947,958	909,325	0.59
Hypochaeris radicata	1,311	1,608,816	1,488,521	0.57
Knautia arvensis	3,608	5,949,258	5,017,209	0.7
Lamium purpureum	1,076	1,578,244	1,467,433	0.68
Leucanthemum vulgare	10,416	13,659,663	9,380,813	0.45
Lolium perenne	2,695	4,091,361	3,682,633	0.68
Lotus corniculatus	465	801,112	771,072	0.83
Malva moschata	978	1,616,874	1,532,383	0.78
Malva sylvestris	1,443	2,265,762	2,041,940	0.71
Matricaria discoidea	2,396	2,841,144	2,473,876	0.52
Medicago lupulina	856	147,000	144,281	0.08
Mimulus guttatus	362	790,364	755,424	1.04
Papaver rhoeas	2,567	3,370,392	2,881,427	0.56
Papaver somniferum	3,716	3,872,190	3,433,193	0.46
Phleum pratense	4,059	3,037,303	2,683,763	0.33
Plantago lanceolata	1,174	1,262,280	1,108,797	0.47
Ranunculus acris	4,352	6,180,375	4,843,466	0.56
Ranunculus repens	10,954	11,943,182	8,548,304	0.39
Reseda luteola	499	175,178	170,678	0.17
Rubus fruticosus	365	643,495	619,240	0.85
Rumex obtusifolius	1,491	2,353,697	2,183,865	0.73
Sambucus nigra	14,915	21,185,013	14,060,492	0.47
Senecio jacobaea	2,201	3,279,071	2,895,207	0.66
Silene vulgaris	1,100	1,528,890	1,425,326	0.65
Stachys sylvatica	1,252	1,491,740	1,410,094	0.56
Trifolium campestre	363	540,421	520,995	0.72
Trifolium repens	1,093	701,115	665,450	0.3
Tripleurospermum maritimum	2,567	4,086,612	3,411,478	0.66
Urtica dioica	1,540	2,028,712	1,835,738	0.6
Veronica agrestis	714	827,108	780,506	0.55
Veronica persica	758	1,487,257	1,392,396	0.92

Appendix 2 Expected vs observed RevMet taxonomic assignments of mock-sample MinION reads

Species	MM1.E	MM1.1	MM1.2	MM2.E	MM2.1	MM2.2	MM3.E	MM3.1	MM3.2	MM4.E	MM4.1	MM4.2	MM5.E	MM5.1	MM5.2	MM6.E	MM6.1	MM6.2
Knautia arvensis	30	32.8	34.6	0	0	0	0	0.1	0	0	0	0	24.8	22.3	24.2	0.2	0.3	0.1
Galium verum	30	16.2	12.9	0	0.2	0.1	0	0	0	0	0.1	0	24.8	9.4	9.9	0.2	0.1	0.1
Crepis capillaris	30	42.8	44.4	0	0	0	0	0.1	0	0	0.1	0	24.8	29.7	27.4	0.2	0.1	0.6
Papaver somniferum	3	1.6	1.2	0.3	0	0	0	0	0	0.1	0.1	0.1	0	0	0	24.8	11.2	10.2
Anagallis arvensis	3	2.5	3.4	0.3	0.4	0.1	0	0.1	0	0	0.1	0	0.2	0.2	0.2	24.8	35.7	36.2
Sambucus nigra	3	3.1	2.8	0.3	0.2	0.8	0	0.1	0.5	83.3	80.8	81.9	0.2	0.2	0.4	24.8	35.1	36.3
Bryonia dioica	0.3	0.1	0	3	1.4	2	4.5	4.4	2.4	0	0	0	0.2	0	0	24.8	16.9	15.9
Ranunculus repens	0.3	0.4	0.3	3	3.2	3	45.2	50.8	54.2	0	0.1	0.1	24.8	28.2	26.8	0	0.1	0.1
Lotus corniculatus	0.3	0	0	3	0.6	1.5	0	0	0	8.3	3.2	3.2	0	0	0	0	0	0
Digitalis purpurea	0	0	0	30	8.8	8.3	4.5	1.7	0.9	0	0	0	0	0	0	0	0	0
Leucanthemum vulgare	0	0	0	30	18.9	17.3	45.2	22.4	23	0	0	0	0	0	0	0	0	0
Stachys sylvatica	0	0	0.1	30	65.1	65.5	0.5	0.9	0.6	8.3	15.4	14.8	0.2	0.8	0.7	0.2	0.4	0.2
Cirsium arvense	0	0	0	0	0	0	0	0	0	0	0	0	0	0.2	0	0	0	0
Centaurea nigra	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Reseda luteola	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Trifolium repens	0	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0.1	0	0	0
Papaver rhoeas	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.2	0.2
Tripleurospermum maritimum	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Rubus fruticosus	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Urtica dioica	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Medicago lupulina	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Trifolium campestre	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ranunculus acris	0	0.1	0.3	0	1	1.3	0	19.3	18.2	0	0.1	0	0	9.1	10.2	0	0	0
Plantago lanceolata	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Holcus lanatus	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Matricaria discoidea	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Appendix 2 continued…

Species	MM1.E	MM1.1	MM1.2	MM2.E	MM2.1	MM2.2	MM3.E	MM3.1	MM3.2	MM4.E	MM4.1	MM4.2	MM5.E	MM5.1	MM5.2	MM6.E	MM6.1	MM6.2
Senecio jacobaea	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Silene vulgaris	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ballota nigra	0	0	0	0	0.2	0	0	0	0	0	0	0	0	0	0	0	0	0
Elymus caninus	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Mimulus guttatus	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Veronica persica	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Veronica agrestis	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Lolium perenne	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Geranium dissectum	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Conium maculatum	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Malva moschata	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Convolvulus arvensis	0	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Achillea millefolium	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bromus commutatus	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Geranium robertianum	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Malva sylvestris	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Phleum pratense	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Hypericum perforatum	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Rumex obtusifolius	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Epilobium hirsutum	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Chaerophyllum temulum	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Lamium purpureum	0	0	0	0	0	0.1	0	0	0	0	0.1	0	0	0	0	0	0	0
Hypochaeris radicata	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Appendix 3 Bee-collected pollen sample information

Bee pollen ID	Bee species	Date collected	Plant captured on	Total DNA (ng)	MinION run	MinION barcode
Am_01	Apis mellifera	01/07/2016	Papaver somniferum	259	1	4
Am_02	Apis mellifera	01/07/2016	Reseda luteola	247	1	7
Am_03	Apis mellifera	01/07/2016	Papaver somniferum	1540	2	4
Am_04	Apis mellifera	01/07/2016	Papaver somniferum	1040	2	7
Am_05	Apis mellifera	01/07/2016	Papaver somniferum	820	2	11
Am_06	Apis mellifera	01/07/2016	Papaver somniferum	720	3	6
Am_07	Apis mellifera	28/06/2016	Reseda luteola	655	3	11
Am_08	Apis mellifera	01/07/2016	Papaver somniferum	625	4	2
Am_09	Apis mellifera	01/07/2016	Papaver somniferum	545	4	7
Bl_01	Bombus lapidarius	20/07/2016	Trifolium repens	274	1	1
Bl_02	Bombus lapidarius	06/07/2016	Trifolium repens	265	1	3
Bl_03	Bombus lapidarius	06/07/2016	Trifolium repens	253	1	5
Bl_04	Bombus lapidarius	06/07/2016	Centaurea nigra	248	1	6
Bl_05	Bombus lapidarius	06/07/2016	Centaurea nigra	232	1	9
Bl_06	Bombus lapidarius	06/07/2016	Centaurea nigra	229	1	10
Bl_07	Bombus lapidarius	06/07/2016	Trifolium repens	227	1	11
Bl_08	Bombus lapidarius	28/06/2016	Trifolium repens	3750	2	1
Bl_09	Bombus lapidarius	28/06/2016	Trifolium repens	2760	2	2
Bl_10	Bombus lapidarius	20/07/2016	Centaurea nigra	1560	2	3
Bl_11	Bombus lapidarius	28/06/2016	Trifolium repens	1350	2	5
Bl_12	Bombus lapidarius	06/07/2016	Trifolium repens	1120	2	6
Bl_13	Bombus lapidarius	20/07/2016	Lotus corniculatus	1010	2	8
Bl_14	Bombus lapidarius	06/07/2016	Trifolium repens	985	2	9
Bl_15	Bombus lapidarius	06/07/2016	Trifolium repens	820	2	12
Bl_16	Bombus lapidarius	28/06/2016	Lotus corniculatus	815	3	1
Bl_17	Bombus lapidarius	01/07/2016	Trifolium repens	735	3	3
Bl_18	Bombus lapidarius	28/06/2016	Reseda luteola	730	3	4
Bl_19	Bombus lapidarius	06/07/2016	Trifolium repens	725	3	5
Bl_20	Bombus lapidarius	06/07/2016	Trifolium repens	700	3	8
Bl_21	Bombus lapidarius	06/07/2016	Trifolium repens	690	3	9
Bl_22	Bombus lapidarius	06/07/2016	Trifolium repens	660	3	10
Bl_23	Bombus lapidarius	06/07/2016	Centaurea nigra	615	4	3
Bl_24	Bombus lapidarius	06/07/2016	Trifolium repens	590	4	4
Bl_25	Bombus lapidarius	28/06/2016	Lotus corniculatus	560	4	6
Bl_26	Bombus lapidarius	01/07/2016	Trifolium repens	505	4	10
Bl_27	Bombus lapidarius	06/07/2016	Centaurea nigra	428	4	12
Bt_lc_01	Bombus terrestris/lucorum complex	06/07/2016	Trifolium repens	271	1	2
Bt_lc_02	Bombus terrestris/lucorum complex	06/07/2016	Centaurea nigra	236	1	8
Bt_lc_03	Bombus terrestris/lucorum complex	01/07/2016	Reseda luteola	191	1	12
Bt_lc_04	Bombus terrestris/lucorum complex	06/07/2016	Trifolium repens	870	2	10
Bt_lc_05	Bombus terrestris/lucorum complex	06/07/2016	Trifolium repens	815	3	2
Bt_lc_06	Bombus terrestris/lucorum complex	01/07/2016	Trifolium repens	705	3	7
Bt_lc_07	Bombus terrestris/lucorum complex	01/07/2016	Reseda luteola	640	3	12
Bt_lc_08	Bombus terrestris/lucorum complex	28/06/2016	Reseda luteola	630	4	1
Bt_lc_09	Bombus terrestris/lucorum complex	06/07/2016	Trifolium repens	580	4	5
Bt_lc_10	Bombus terrestris/lucorum complex	28/06/2016	Reseda luteola	515	4	8
Bt_lc_11	Bombus terrestris/lucorum complex	28/06/2016	Trifolium repens	515	4	9
Bt_lc_12	Bombus terrestris/lucorum complex	28/06/2016	Reseda luteola	496	4	11

Appendix 4 RevMet taxonomic assignments for Bee-collected pollen samples

Bee pollen ID	*Achillea millefolium*	*Ballota nigra*	*Centaurea nigra*	*Elymus caninus*	*Holcus lanatus*	*Hypochaeris radicata*	*Lotus corniculatus*	*Papaver rhoeas*	*Papaver somniferum*	*Reseda luteola*	*Rubus fruticosus*	*Trifolium campestre*	*Trifolium repens*	*Urtica dioica*
Am_01	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.6	96.9	0.0	0.0	0.0	1.6	0.0
Am_02	0.0	0.0	1.9	0.0	0.0	0.0	0.0	0.0	0.0	93.2	0.0	0.0	2.9	1.9
Am_03	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.7	97.3	0.0	0.0	0.0	0.0	0.0
Am_04	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.9	97.1	0.0	0.0	0.0	0.0	0.0
Am_05	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.5	98.5	0.0	0.0	0.0	0.0	0.0
Am_06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.2	96.8	0.0	0.0	0.0	0.0	0.0
Am_07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0
Am_08	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.2	97.8	0.0	0.0	0.0	0.0	0.0
Am_09	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6.5	93.5	0.0	0.0	0.0	0.0	0.0
Bl_01	0.0	0.0	83.2	0.0	0.0	0.0	4.3	0.0	0.0	0.0	0.0	0.0	12.5	0.0
Bl_02	0.0	0.0	7.9	1.2	0.0	0.0	13.5	0.0	0.0	0.0	0.0	1.5	75.9	0.0
Bl_03	0.0	0.0	1.8	1.1	0.0	2.4	2.0	0.0	0.0	0.0	0.0	3.5	89.3	0.0
Bl_04	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.8	97.2	0.0
Bl_05	0.0	0.0	0.0	1.7	0.0	0.0	93.4	0.0	0.0	0.0	0.0	0.0	4.9	0.0
Bl_06	1.4	0.0	97.3	0.0	0.0	0.0	1.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Bl_07	0.0	0.0	14.6	1.4	0.0	0.0	1.2	0.0	0.0	0.0	0.0	1.2	81.7	0.0
Bl_08	0.0	0.0	0.0	0.0	0.0	0.0	96.6	0.0	0.0	0.0	0.0	0.0	3.4	0.0
Bl_09	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Bl_10	0.0	0.0	17.1	0.0	0.0	0.0	80.6	0.0	0.0	0.0	0.0	0.0	2.3	0.0
Bl_11	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Bl_12	0.0	0.0	0.0	0.0	0.0	0.0	72.8	0.0	0.0	0.0	0.0	0.0	27.2	0.0
Bl_13	0.0	0.0	37.4	0.0	0.0	0.0	62.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Bl_14	0.0	0.0	0.0	1.1	0.0	1.1	33.1	0.0	0.0	0.0	0.0	1.6	63.1	0.0
Bl_15	0.0	0.0	0.0	0.0	0.0	0.0	18.8	0.0	0.0	0.0	0.0	2.3	78.9	0.0
Bl_16	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Bl_17	0.0	0.0	0.0	0.0	0.0	0.0	33.0	0.0	0.0	0.0	0.0	1.8	65.2	0.0
Bl_18	0.0	0.0	0.0	0.0	1.4	0.0	75.6	0.0	2.1	20.9	0.0	0.0	0.0	0.0
Bl_19	0.0	0.0	0.0	1.1	0.0	0.0	25.8	0.0	0.0	0.0	0.0	1.4	71.7	0.0
Bl_20	0.0	0.0	2.8	0.0	0.0	0.0	23.1	0.0	0.0	0.0	0.0	1.7	72.3	0.0
Bl_21	0.0	0.0	0.0	0.0	0.0	0.0	55.3	0.0	0.0	0.0	0.0	1.7	43.0	0.0
Bl_22	0.0	0.0	0.0	0.0	0.0	0.0	84.6	0.0	0.0	0.0	0.0	0.0	15.4	0.0
Bl_23	0.0	0.0	4.3	0.0	0.0	0.0	95.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Bl_24	0.0	0.0	0.0	0.0	0.0	0.0	12.0	0.0	0.0	0.0	0.0	1.6	86.4	0.0
Bl_25	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Bl_26	0.0	0.0	0.0	0.0	0.0	0.0	95.5	0.0	0.0	0.0	0.0	0.0	4.5	0.0
Bl_27	0.0	0.0	0.0	0.0	0.0	0.0	43.4	0.0	0.0	0.0	0.0	1.8	54.7	0.0
Bt_lc_01	0.0	1.7	1.7	0.0	1.7	0.0	0.0	0.0	1.7	0.0	72.4	3.4	13.8	3.4
Bt_lc_02	1.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6.4	1.9	90.5	0.0
Bt_lc_03	0.0	0.0	3.8	0.0	0.0	0.0	19.0	2.5	72.2	0.0	0.0	0.0	2.5	0.0
Bt_lc_04	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Bt_lc_05	0.0	0.0	0.0	0.0	0.0	0.0	4.3	0.0	0.0	0.0	0.0	1.9	93.8	0.0
Bt_lc_06	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.2	98.8	0.0
Bt_lc_07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.1	76.4	20.5	0.0	0.0	1.0	0.0
Bt_lc_08	0.0	0.0	0.0	0.0	0.0	0.0	80.8	0.0	0.0	0.0	0.0	0.0	19.2	0.0
Bt_lc_09	0.0	0.0	0.0	1.6	0.0	0.0	50.0	0.0	0.0	0.0	0.0	1.2	47.2	0.0
Bt_lc_10	0.0	0.0	0.0	0.0	0.0	0.0	1.9	0.0	0.0	95.3	0.0	0.0	2.7	0.0
Bt_lc_11	0.0	0.0	0.0	0.0	0.0	0.0	17.9	0.0	0.0	0.0	0.0	1.8	80.4	0.0
Bt_lc_12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0

Plant species on which bee was foraging when collected