# Application of Wavelet-based Denoising to Improve the Accuracy of Nanopore Sequencing Data

A thesis submitted to the
College of Graduate and Postdoctoral Studies
in partial pulfillment of the requirements
for the degree of Master of Science
in the Department of Computer Science
University of Saskatchewan
Saskatoon

By
Saiyida Noor Fatima Topping

# Permission to Use

# Disclaimer

Reference in this thesis to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement, recommendation, or favoring by the University of Saskatchewan. The views and opinions of the author expressed herein do not state or reflect those of the University of Saskatchewan, and shall not be used for advertising or product endorsement purposes.

Requests for permission to copy or to make other uses of materials in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
176 Thorvaldson Building, 110 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan S7N 5C9 Canada

OR

Dean
College of Graduate and Postdoctoral Studies
University of Saskatchewan
116 Thorvaldson Building, 110 Science Place
Saskatoon, Saskatchewan S7N 5C9 Canada

# Abstract

DNA sequencing methods in biology are divided into three generations based on their time of invention and technology used. First generation sequencing technologies introduced in the 1970s sequenced short strands of DNA, with the longest strand ranging from 300-1000 base pairs in the Sanger method. Second generation technologies improved on the first generation by being high throughput, scalable and parallel. After successful genome assemblies of small and large organisms using first and second generation sequencing methods, the last two decades brought about third generation sequencing technologies. Third generation sequencing technologies focus on sequencing single nucleotide molecules and produce real-time, high-throughput basecalls and are scalable, low cost and portable. Nanopore sequencing is a third generation sequencing technology that works by measuring the change in electric current in an ionic membrane as a DNA strand passes through a nanopore embedded in the membrane. A major limitation that has prevented mass adoption of nanopore sequencing commercially is its lower accuracy compared to second generation sequencing technologies. The aim in this project was to improve the accuracy of nanopore sequencing by reducing noise in the nanopore signal. Wavelets were used to decompose the nanopore signal, remove noise and then reconstruct the signal. The modified signal was used for training a new basecalling model. It was observed that a significant difference in basecall quality can be achieved between the default model used by Oxford Nanopore Technologies's Guppy basecaller and our custom denoised model in terms of mean percentage identity. An increase of 5.3% was achieved in mean percentage identity while maintaining the mean read quality of basecalls for *Bacteriophage lambda* dataset. Both mean percentage identity and mean read quality for the custom model were overall more consistent with lesser low scoring outliers. Haar wavelet was demonstrated as the most suitable wavelet candidate with level of decomposition and threshold values 4 and 0.04 respectively for denoising nanopore sequencing data. Results were validated by training and testing with and without wavelet denoising on three existing nanopore datasets.

# Acknowledgements

I would like to express my gratitude to my supervisors Dr. Matthew Links and Dr. Kevin Stanley for their supervision and support during my research. I could not have accomplished this project without their knowledge, guidance and encouragement. I am also grateful to my advisory committee for their invaluable feedback, suggestions and corrections.

Additionally, I would like to thank my family and friends for their continuous support and love, without which I could not be where I am today.

This thesis is dedicated to my husband William and my daughter Zaina.

# Contents

# List of Tables

# List of Figures

# List of Abbreviations

| | |
|---|---|
| bp | Base pair |
| CTC | Connectionist temporal classification |
| dB | decibels |
| DNA | Deoxyribonucleic acid |
| FFT | Fast fourier transform |
| HMM | Hidden markov model |
| NCBI | National Center for Biotechnology Information |
| NHGRI | National Human Genome Research Institute |
| ONT | Oxford Nanopore Technologies |
| PSD | Power spectral density |
| RNA | Ribonucleic acid |
| RNN | Recurrent neural network |
| SNR | Signal to noise ratio |

# 1 Introduction

## 1.1 DNA Sequencing

Deoxyribonucleic acid, commonly referred to as DNA, is a double helical structure of polynucleotide chains made up of four nucleotide bases: adenine (A), guanine (G), cytosine (C) and thymine (T). DNA carries genetic information for all living beings based on the sequence of nucleotide bases. It stores the information necessary for all protein creation in living organisms for development, growth and reproduction. Determining the sequence of the nucleotide bases in a DNA strand is called DNA sequencing. DNA sequencing is the most effective method for studying genes on a molecular level, as it enables scientists to study the encoded information in nucleotide sequences and its resulting gene expression and phenotype. Since the discovery of DNA as a genetic material in 1944 [1], and its double helix structure in 1953 [2] [3], biologists have been working to devise methods for sequencing DNA efficiently and accurately. The evolution of DNA sequencing technologies is categorized into three generations. First generation sequencing started in 1970s with the advent of the Maxam and Gilbert method [4], followed by the Sanger method [5]. Second generation sequencing technologies improved on the first generation technologies and lowered the cost and time of sequencing by making sequencing scalable and parallel. Third generation sequencing technologies introduced real-time sequencing [6] with longer reads compared to first and second generation sequencing [7]. PacBio SMRT sequencing [8] and nanopore sequencing [9] are the leading technologies in third generation sequencing. Second generation sequencing technologies such as Illumina [10] are most commonly used in academic and commercial sequencing applications, while third generation sequencing technologies face a challenge in mass adoption due to comparatively lower accuracy. Errors that reduce the accuracy of a DNA sequence include insertions, deletions and substitutions.

1. Insertion errors occur when a nucleotide base is introduced into the basecalled sequence at a location where it does not exist in the reference sequence.

2. Deletion errors occur when a nucleotide base is skipped while basecalling.

3. Substitution errors occur when a nucleotide base is called incorrectly as a different base.

## 1.2 Nanopore Sequencing

Nanopore sequencing is a third generation sequencing technology. It works by measuring the change in electric current while the DNA nucleotides pass through a protein nanopore embedded in an electrically charged lipid bi-layer membrane. As DNA naturally carries a negative charge, a DNA strand is passed through the nanopore by applying a voltage across the pore. The idea of nanopore sequencing was first described by David Dreamer and Daniel Branton in 1996 [11]. They discovered that a single DNA strand can be passed through a protein nanopore, with each nucleotide base in DNA affecting the ionic conductivity of the pore as it passes. In 2001, Hagan Baley published the description of a nanopore sensor's workings for DNA sequencing [9]. Once it had been established that it was possible to sequence DNA by measuring current fluctuations, researchers looked into controlling the speed of DNA translocation through the pore so measurements can be taken periodically. Motor protein enzymes, for example helicases and polymerases, are used as translocation speed controllers for single strand DNA passing through a nanopore. Electric current fluctuation measurements correspond to the sequence of bases in the DNA strand. The relationship between current fluctuation and nucleotide bases is derived using machine learning methods and algorithms. In 2005 Hagan Baley, along with some of his colleagues, founded the organization known as Oxford Nanopore Technologies Limited. Oxford Nanopore Technologies (`ONT`) has been producing nanopore sequencing devices for commercial and academic use since 2014 [11] and remains the sole provider for nanopore sequencing devices [12]. Nanopore sequencing devices by Oxford Nanopore Technologies use "flow cells" consisting of nanopores embedded in electro-resistant membranes. The electric current passing through a nanopore is measured by the a sensor chip connected with that nanopore's corresponding electrode. Change in electric current caused by the flow of DNA molecules through the nanopore produces the raw nanopore signal used for basecalling.

### 1.2.1 Advantages of Nanopore Sequencing

Nanopore sequencing devices provide real-time detection of single DNA and RNA molecules with read length ranging between 500bp to 2.3Mbp with an average of 5kbp, which is a significant improvement over first and second generation sequencing methods' read lengths. First and second generation sequencing methods produce read lengths well below 1 kilobases. Longer read length allows for greater overlap between reads, increased possibility of spanning repeat regions and increases confidence in the basecalling accuracy. An interesting advantage of nanopore sequencing technology is that the accuracy of one basecall does not depend on the previous basecalls for a single strand of DNA [13]. `ONT MinION` devices provide real-time access to individual reads which can be analyzed as soon as they are produced [14]. Nanopore data improves the *de novo* assembly of genomes and offers accurate resolution of structural variants [15]. First and second generation sequencing technologies use DNA amplification methods like the Polymerase Chain Reaction (PCR) [16] to create multiple copies of DNA strands in a DNA sample. Amplification can introduce errors and bias in the

DNA sample as compared to the original sample (prior to amplification) [11]. Nanopore sequencing works without amplification and thus requires minimal sample preparation chemistry, lowering the cost of nanopore sequencing to $<$ USD1000 for a mammalian genome [13]. Nanopore sequencing is a preferred option for some clinical applications due to the following advantages:

1. Long reads: nanopore sequencing provides a read length ranging between 500bp to 2.3Mbp [17] which is the longest of any commercial sequencing technology. Most commonly, the read length in nanopore sequencing averages at 5kbp. Longer reads improve the confidence in basecalling accuracy due to the increased chance of spanning repeat regions and greater overlapping regions.

2. Portability and small size: An `ONT` `MinION` device is 4 inches long and powered over a USB connection. It makes DNA sequencing accessible to a wide range of people interested in genomics without the overhead of requiring an elaborate molecular biology laboratory [14].

3. Low cost and time: nanopore sequencing does not require cloning and amplification steps which results in minimal sample preparation steps [13]. This aspect of nanopore sequencing significantly decreases the cost and effort associated with DNA sequencing. The cost of purchasing a nanopore `MinION` device is USD1000 for a basic package which includes the device, a flow cell, and chemical reagents required to conduct a sequencing experiment. An enhanced `MinION` package that offers training and multiple flow cells costs USD3300. In comparison, the Sanger method from first generation sequencing and any of the second generation sequencing methods are expensive, both in terms of the instruments and chemical reagents required for sequencing [18].

4. Ease of use: Oxford Nanopore Technologies' mission is "to enable the analysis of anything, by anyone and anywhere". Nanopore sequencing devices are easy to use and make it possible for anyone to sequence DNA anywhere and have great prospects of being used for clinical sequencing needs [19].

Improvements in nanopore sequencing have increased the opportunities for its various applications. Nanopore sequencing has great potential in the study of rare and genetic diseases, gene mutations, and single nucleotide polymorphism research [20] due to its portability, low cost and realtime basecalling ability. A feature of nanopore sequencing is that in addition to generating a sequence, nanopore devices also provide us with the raw electric signal obtained from the nanopore. Analysis of the raw signal carries many opportunities for studying the characteristics of DNA and protein nanopores [14]. Access to the raw nanopore signal provides researchers with the option to create their own basecallers, genome assembly tools and pre-processing signal and post-processing (polishing) algorithms [14]. Similar to the ability of examining Sanger trace files, researchers using nanopore sequencing can refer to the raw signal for clues if they do not have confidence in the basecalled reads. Access to raw nanopore signal also provides the opportunity to apply signal analysis techniques to the data prior to basecalling, which is explored in this thesis.

### 1.2.2 Challenges of Nanopore Sequencing

Due to their relatively higher error rate, third generation sequencing technologies fall behind when compared with second generation sequencing technologies in their adoption and use [10]. For example, Illumina from the second generation has an impressive error frequency of $10^{-3}$ [21], making the basecalled sequences 99.9% accurate in alignment, whereas, sequence alignment accuracy for nanopore sequencing is typically between 94-97% [22].

The reasons behind lower accuracy in nanopore sequencing include noise in the signal, non-deterministic timing of when a DNA molecule is halted by the control enzyme, and potential systematic channel errors [23]. Other sources of errors can include training datasets, basecalling models, and algorithms. Developments in third generation sequencing technologies are overcoming many of these challenges, but nanopore sequencing remains less preferred when compared with second generation technologies [17].

Another known limitation of nanopore sequencing is the inaccurate sequencing of homopolymers. Homopolymers are regions in a DNA sequence containing the same nucleotide consecutively. Incorrect sequencing of homopolymers is a challenge faced by other sequencing technologies [22] as well. Insertion and deletion errors are common in homopolyer regions sequenced by Illumina, Ion Torrent and Roche 454 leading to incorrect length of the homopolymer region [24]. The number of deletion errors for homopolymer sequences can increase up to 2.6 times that of a regular sequence according to a study of clinical genome sequencing of patients with congenital abnormalities [25]. A larger rate of deletion errors for homopolyers leads to the homopolymer region not being accurately represented in the basecalled sequence. This high rate of inaccurate homopolymer sequencing occurs due to the fact that the signal may not change while a homopolymer passes through the nanopore [26].

Despite of the limitations in nanopore sequencing, hardware and software have improved, resulting in overall nanopore sequencing technology's evolution and improvement [22]. Hardware improvements include improvements in protein nanopores [27] and control over DNA translocation speed [28]. Software improvements include improvements in basecalling methods [29] and the introduction of choices of basecallers [30].

Most random errors can be eliminated in a consensus sequence, which is generated by overlapping multiple sequences from the same location in the genome to mitigate against any variations and inaccuracies in nucleotides, and requires that each sequence in the genome must be basecalled more than once. There are tools available for genome assembly and consensus calling that polish the sequence after basecalling for example `Nanocorrect`, `Nanopolish` [31], `Canu` [32]. Improving individual read accuracy can reduce the dependence on consensus sequence and post-processing of the basecalled sequence, saving time and cost.

For nanopore sequencing to become the premier sequencing method, the single-read accuracy needs to improve to match that of second generation sequencing methods. One way to improve the accuracy is to reduce the amount of noise in the electric signal generated by nanopore devices. In this thesis work it is hypothesised that removing noise from the signal and creating new models for basecalling will increase the accuracy of basecalls, making nanopore sequencing appropriate for more diverse applications. The ionic current across a nanopore changes when a nucleotide passes from within the nanopore and remains stable otherwise. Within the nanopore there can be multiple nucleotides at a time, depending on the size of the nanopore. Nucleotide molecules ratcheting to the next position within the pore also causes fluctuations in the current across the nanopore. A nanopore signal depicts periods of time when the current appears stable as well as the short time periods when it varies. In addition to the actual signal information, a nanopore signal also captures errors in measurements due to experiment setup, biochemical agents or the design and type of nanopore itself. The causes and types of noise in a nanopore signal are explained in Section 2.2.2.

Prior to this thesis work there has been no published demonstration of noise removal; however, wavelets have been studied before for electrochemical noise analysis [33]. There is evidence in other domains that removing noise from a signal before analysis produces more accurate results [34].

This thesis demonstrates the extent to which basecalling accuracy increases using wavelet denoising on a nanopore signal. Wavelet analysis was used to decompose the nanopore signal, remove high frequency noise, and recompose the signal. The resulting signal was then used to train the basecalling model and used the new model for basecalling. Signal processing techniques were used to determine the noise band in the nanopore signal and signal-to-noise ratio was used as a metric to narrow the list of possible parameter values for wavelet analysis. Three experimental datasets were explored in this thesis, consisting of small to large genomes including *Bacteriophage lambda*, cattle, and a multi-species dataset including *E. coli*, yeast and human genomes. The datasets are outlined in detail in Section 4.1.

In this thesis it is demonstrated that reducing noise in a nanopore signal and creating custom basecalling models with denoised signal data led to a higher read accuracy and improved read quality. Described is an increase in basecalling accuracy when a custom model based on denoised training data was used with raw testing data. `NanoStat` [35] was used to evaluate the basecalls and calculate mean percentage identity and mean read quality for each sequence. Increase in read accuracy of nanopore sequencing opens doors for additional research and opportunities for commercial and academic use of nanopore sequencing devices.

# 2 Background

## 2.1 Basecalling

Nanopore sequencing technology commonly employs machine learning methods to derive DNA sequences from an electric signal produced by the nanopore devices. The process of interpretation of experimental observations into a sequence of nucleotide bases in DNA is known as basecalling. Machine learning models that are commonly used in basecalling nanopore sequencing include:

1. Hidden Markov Model: HMMs have been used to basecall nanopore sequencing data statistically in basecallers such as `Nanocall` [36].

2. Recurrent Neural Network: RNNs are a popular choice for basecalling nanopore sequencing data in the more recent basecallers and have improved in performance over the years [30].

3. Connectionist Temporal Classification: Temporal Classification (CTC) [37] generates a probability distribution of a single base (as opposed to k-mers) for each position while eliminating the need to preprocess the signal. A flip-flop model follows the CTC style for calling the sequence base by base using an RNN.

Table 2.1 lists a few of the many available basecalling tools. Figure 2.1 shows a timeline of the commonly used basecallers. The active period for each basecaller was determined based on the basecaller's release date and most recent updates. With developments in the nanopore technology and machine learning methods, the tools provide a variety of methods for basecalling with varying performances [30]. For this thesis, `Guppy` basecaller by `ONT` was used, both with and without wavelet denoising. `Guppy` was chosen because of its current popularity and stability, overall performance and accuracy [30] in basecalling along with its ability to easily plug in any custom model during runtime.

Sequences generated after basecalling can be analyzed for their accuracy and correctness to determine the best basecalling methods and techniques. Accuracy of sequences produced by a basecaller was measured by comparing it against a known reference sequence and looking for insertions, deletions, mismatches and matches across the two sequences. Existing tools in the open-source community provide implementations of widely used bioinformatics algorithms for comparing sequences. Over the course of this thesis `minimap2` [41] and `NanoPack` [35] were used as the major tools for assessing basecalling accuracy and analysis. The `MapQ` score by `minimap2` is based on the commonly used `Phred` quality scale for error analysis in sequences.

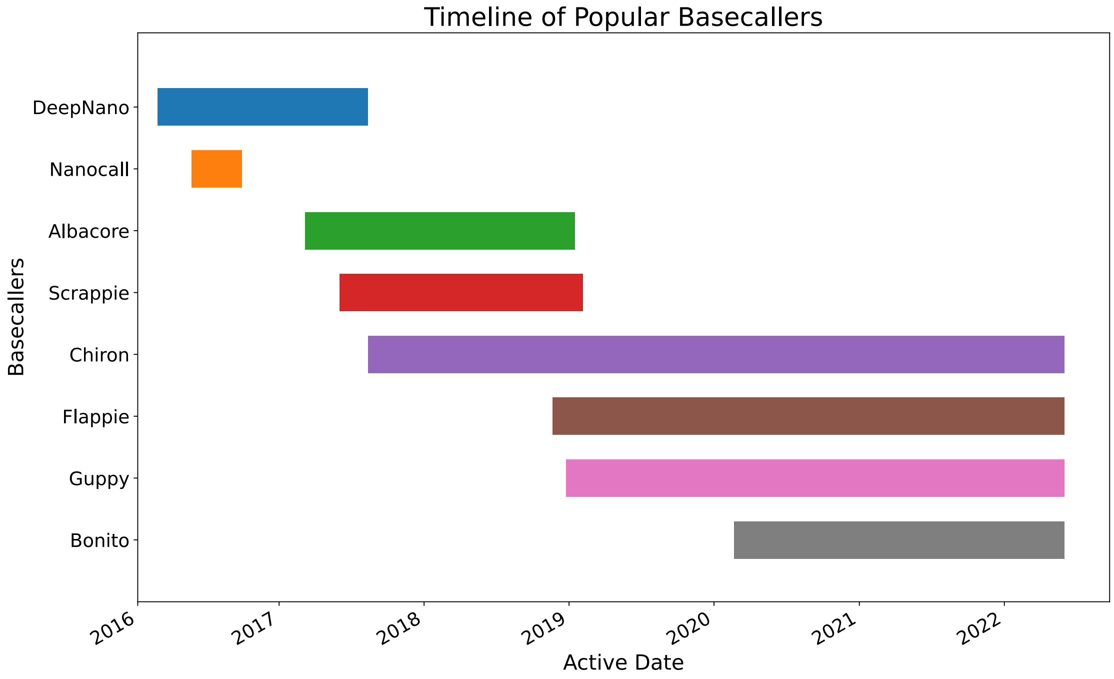


**Figure 2.1:** Active period of commonly used basecallers from Table 2.1

The `Phred` quality score is logarithmically related to error probabilities. For example, a `Phred` score of 10 corresponds to 99% accurate basecalls or a probability of 1 in 10 incorrect basecall. Similarly a `Phred` score of 20, 30 or 40 corresponds to 99.9%, 99.99% or 99.999% accuracy respectively. `MapQ` score by `minimap2` is generated on a per unique read basis and ranges from 0-60. Any read that aligns well to more than one reference sequence is given a `MapQ` score of 0 and is considered multi-mapped. Highly accurate alignments are given a score of 60. We also referred to the online `Blast` implementation [42] on NCBI at times for our analysis.

| Name | Description | Active Period | Author | Reference |
|---|---|---|---|---|
| `Albacore` | `Albacore` is a general purpose basecaller that runs on CPUs. | v1.2.4 03/07/2017 to v2.3.4 15/01/2019 | Oxford Nanopore Technologies | [30] |
| `Guppy` | `Guppy` is a neural network based basecaller that frequently outperforms other basecallers in terms of accuracy and performance. It is similar to `Albacore` but can utilize GPUs for increasing basecalling speed. | v2.1.3 24/12/2018 to Present | Oxford Nanopore Technologies | [30] |
| `Scrappie` | `Scrappie` is an open-source basecaller that Oxford Nanopore Technologies has used to demonstrate the basecalling technology to the community. It is often the first basecaller to incorporate new features that are later added to other basecallers. | 0.2.3 Feb 6, 2017 to 1.4.2 Apr 2, 2019 | Oxford Nanopore Technologies | [30] |
| `Flappie` | `Flappie` replaced `Scrappie` as the open-source basecaller to demonstrate basecalling technology. | v0.1.0 Nov 21, 2018 to Present | Oxford Nanopore Technologies | [30] |
| `Bonito` | `Bonito` is a `PyTorch` based open-source basecaller that explores alternative technologies to basecalling. | v0.0.5 Feb 20, 2020 to Present | Oxford Nanopore Technologies | [38] |
| `Nanocall` | `Nanocall` is an open-source basecaller that works offline. | v0.5.13 May 26, 2016 to v0.7.4 Sep 29, 2016 | Matei David *et al.* | [36] |
| `Chiron` | `Chiron` is a basecaller built on `Tensorflow` and uses CNNs, RNNs and CTC for basecalling. | 0.1 Aug 13, 2017 to Present | Haotian Teng *et al.* | [39] |
| `DeepNano` | `DeepNano` is an open-source basecaller based on deep RNNs. | March 14, 2016 to August 13, 2017 | Vladimír Boža *et al.* | [40] |

**Table 2.1:** Basecalling tools for nanopore sequencing published by Oxford Nanopore Technologies and the bioinformatics research community.

## 2.2 Nanopore Signal

### 2.2.1 Signal

A signal is a mathematical function that represents information in the form of a wave. A nanopore signal represents the change in ionic current across a protein nanopore as a single DNA strand passes through it. Each nucleotide base (A, T, G, C) in a DNA strand interferes with the ionic current across a nanopore in a unique way, causing distinct fluctuation in the current. These fluctuations are used to detect the nucleotide base passing through the pore at any one point in time. The current remains stable during the stationary period for the strand. Movement of the DNA strand can be detected from the changes in the signal amplitude. A few key terms related to signal analysis that were used in this thesis are:

- Time series: A time series in mathematics is a series of data points presented in intervals of time.
- Time domain: Time domain refers to presenting the changes in signal with respect to time.
- Frequency: Frequency of a signal represents the number of waves in a signal per unit time.
- Frequency domain: Frequency domain refers to presenting the amount of signal with respect to frequency bands.
- Amplitude: Amplitude of a wave represents the distance between the peak of a wave or the trough of a wave to the center line.
- Trigonometric functions: The mathematical functions sine, cosine and tangent, and their respective inverses cosecant, secant and cotangent are collectively referred to as trigonometric functions.
- Signal coefficients: A signal can be deconstructed into a series of values in the frequency domain called signal coefficients. The coefficients can be of two types: approximation coefficients with low frequency information of the signal, and detail coefficients with high-frequency information of the signal.
- Power spectral density (PSD): Power spectral density of a signal represents the power per unit frequency in a signal.
- Noise: Noise is the interference or disturbance in an electric signal that is not desired.
- Signal to noise ratio (SNR): Signal to noise ratio is commonly used to express the relationship between an electric signal and background noise.

Raw nanopore signal generated by `MinION`, a nanopore sequencing device by `ONT`, was used for all experiments in this thesis work.

### 2.2.2 Noise in the Nanopore Signal

One of the contributors to inaccuracy in basecalls is the presence of noise in the electric signal produced by the nanopore devices. The types and sources of noise in the nanopore signal can vary based on the hardware, type of the nanopore protein used, and experiment setup. `ONT` uses two types of nanopore in their devices: biological nanopores that exist in nature and solid-state nanopores that are created in synthetic membranes. Early generations of `ONT` devices used biological nanopores created using programmed bacteria while later generations of `ONT` devices use solid-state nanopores. A study that conducted a comparison between biological and solid-state nanopores shows a higher SNR in solid-state nanopores [43], specifically silicon nitride nanopores. Solid-state nanopores are the preferred protein nanopore for nanopore sequencing. The same study also showed an increase in SNR when the DNA translocation speed is slowed in the nanopore. Another study [44] noted that in solid-state nanopores, the noise in signal can vary between different pores. A higher resistance across nanopores contributes to increased low-frequency noise in signal. These factors are not considered in this thesis as they are determined by the hardware, biochemistry and manufacturing of the device. In this thesis, the data generated by nanopore devices was used and signal processing techniques were applied to that data. The physical devices and experimental setup remained fixed throughout the experiments.

Noise in a nanopore signal is commonly modeled as being from the following types:

1. $1/f$ noise : Known as pink noise, $1/f$ noise is a signal component that has a spectral density inversely proportional to the frequency of the signal. While no single source of $1/f$ noise generation in nanopore sequencing has been confirmed, $1/f$ noise is expected from many different biochemical sources. Changing the nanopore materials, pH and membrane surface charge density can all lead to a change in $1/f$ noise [45].

2. White noise: White noise is a signal component that has the same amplitude and intensity throughout the signal. In other words, the power spectral density of white noise does not depend on the signal's frequency. It is caused by a combination of thermal fluctuations and potential barriers in the nanopore devices [45]. Decreasing the pore size or conductance can reduce white noise in the nanopore system.

3. Dielectric noise: Dielectric noise is the noise generated by thermodynamic fluctuations in dielectric materials. Existence of a leakage current that is not exclusive to the nanopore is a cause of this noise in the nanopore devices. Reducing the capacitance and using a stacked dielectric structure in the nanopore devices can reduce dielectric noise in the system [45].

4. Amplifier noise: Amplifier noise is generated by the interaction of capacitance with thermal noise at the amplifier input and typically is a high frequency noise. It can be reduced by reducing the capacitance of the system or by using low capacitance amplifiers [45].

Each type of noise in the nanopore signal may be distributed at different frequencies; however, a granular

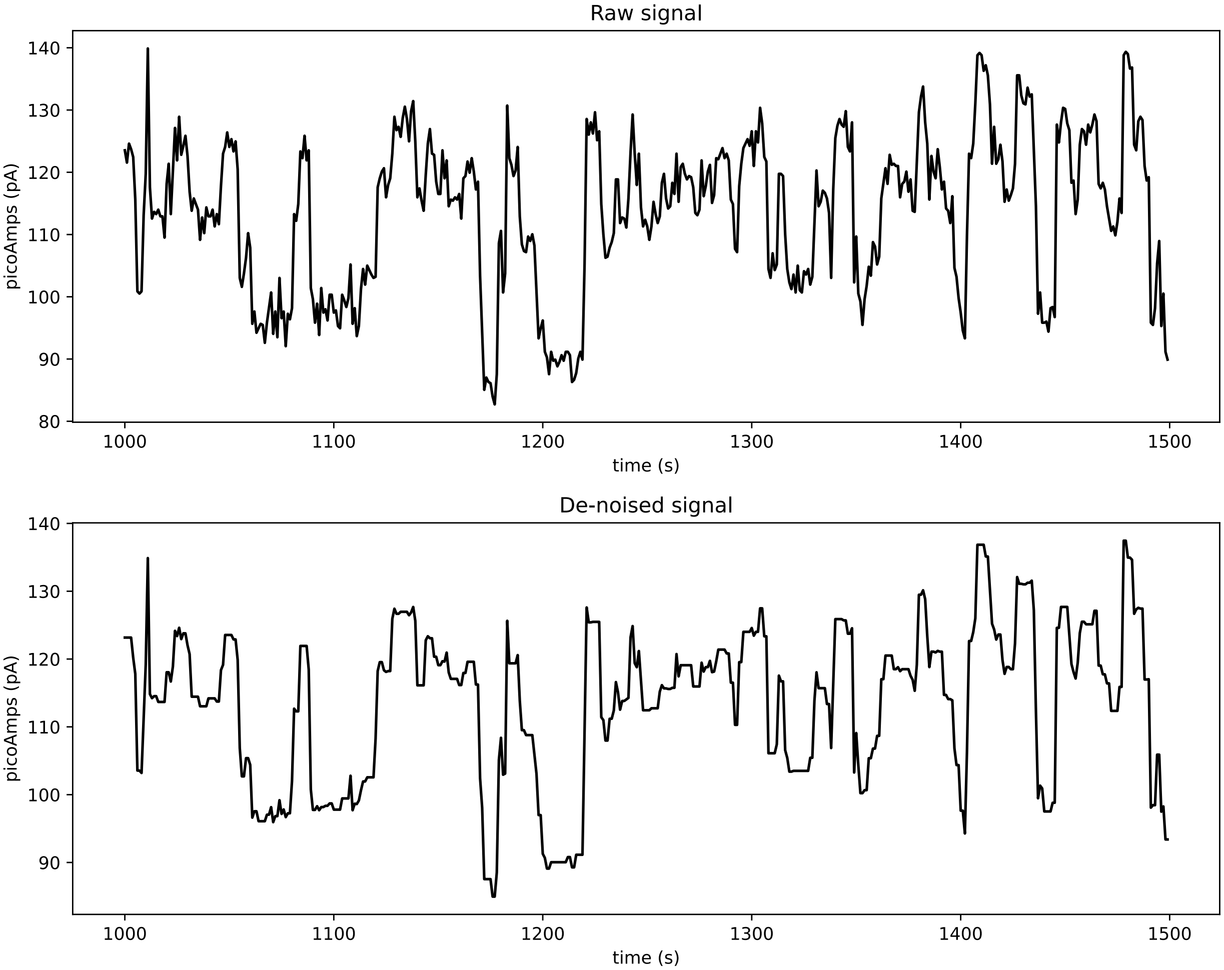


**Figure 2.2:** Comparison of raw signal with its denoised version. This sample signal was denoised using the `haar` wavelet, decomposition level = 4 and threshold = 0.04

view of the types of noise present in the datasets involved in this thesis work is not available and all of the above types of noise is collectively referred to as "noise" from now on.

Figure 2.2 shows a comparison of a nanopore signal before and after denoising using the `haar` wavelet, $decomposition\quad level = 4$ and $threshold = 0.04$. The detailed process of denoising the signal and choice of wavelet and parameters has been explained in Chapter 4. Notice the processed signal contains less noise at the peaks and troughs of the signal, but still conforms to the nanopore signal shape, ensuring that important signal detail was not removed. The processed signal also shows the events occurring in the signal much more obviously in a visual observation than the raw signal. We believe that this noise reduction in the nanopore signal will lead to improved clarity in the detection of events, thus leading to better basecalling. The parameters selected to denoise the signal in Figure 2.2 are explained in Chapter 4 in detail.

### 2.2.3 Signal to Noise Ratio

Signal to noise ratio (SNR) is a commonly used measure to express the relationship between a desired electric signal and background noise [46]. It is the ratio of signal power to noise power in a signal, expressed in decibels. Improvements have been made in the selection and design of nanopore proteins such as reducing the thickness of membrane, resulting in an increase in the magnitude of signal [27]. SNR is higher in pores that are smaller in diameter (1.2nm) compared to larger pores [47]. Solid-state nanopores have been proven to generate signal with higher SNR than biological protein nanopores [27]. However, some amount of background noise is inevitable in any electrical measurement. The change in SNR was used to determine the optimal noise reduction methods and parameters for this thesis. A higher SNR after denoising indicates the reduction of noise while preserving signal information. No change in SNR or a decrease in SNR after denoising indicates a potential loss of important information from the signal. Evaluating the change in SNR after filtering can serve as an indicator of whether the signal has been over-filtered or under-filtered.

## 2.3 Signal Processing

### 2.3.1 Fourier Analysis

To understand the wavelet analysis and signal processing for this project, a brief understanding of the Fourier transform is required. The Fourier Theorem introduced by J.B. Fourier is a mathematical theorem that proves that any arbitrary signal can be represented as a sum of sine and cosine functions. Fourier analysis [48] implements the Fourier Theorem and breaks down a signal into sinusoidal signals, representing it as a sum of trigonometric functions. The Fourier transform is a mathematical transform widely used in signal processing that decomposes a signal based in time and reconstructs it as a signal based in frequency. This makes it possible for the signal to be analysed in a different domain than its original domain. Breaking down a signal into its composing wave forms enables the analysis of the signal on a granular scale, with the ability to adjust power spectral densities of each waveform and analyse the patterns in the composition which may not be visible from looking at the original signal. Fourier analysis consists of a Fourier transform to decompose the signal and an inverse Fourier transform to then reconstruct the signal using frequency components. Fourier analysis's ability to determine components of a signal is used in noise removal, compression and identifying patterns which are all useful properties applied in digital signal processing [49]. Fourier analysis lacks the ability to provide localized detail about the signal in the time domain and cannot provide detail in any one specific portion of the signal. A window function is sometimes used with the Fourier transform to focus on smaller portions of the signal during analysis and then shifted along the axis to other portions of the signal. This version of the Fourier transform is known as Windowed Fourier Transform [50]. A Windowed Fourier Transform offers localization in both time and frequency domains.

### 2.3.2 Wavelet Analysis

A wavelet is a signal of limited duration (in time) that starts and ends with an amplitude of zero, and has an increasing and decreasing amplitude in between the start and end point in time. The area under the curve of a wavelet signal adds up to zero. Wavelets can also be defined as a family of functions derived from a single mother wavelet. Decomposing a signal into a set of wavelet functions to reveal hidden patterns is known as the wavelet transform [51]. There are seven discrete families of wavelets that were considered for this project. A list of wavelet families and their example applications are provided in Table 2.2. Figure 2.3 shows five of the most commonly known wavelets that are used in a discrete wavelet transform.

| Family | Symmetry | Example Application |
|---|---|---|
| Haar (`haar`) | Asymmetric | Network traffic prediction [52] |
| Daubechies (db) | Asymmetric | Speech signal processing [53] |
| Symlets (sym) | Near symmetric | Partial discharge signals denoising [54] |
| Coiflets (coif) | Near symmetric | Partial discharge signals denoising [54] |
| Bioorthogonal (bior) | Symmetric | Image compression [55] |
| Reverse Bioorthogonal (rbio) | Symmetric | Detection of QRS complexes in ECG signal [56] |
| Discrete approximation of Meyer (dmey) | Symmetric | Wavelength detection accuracy [57] |

**Table 2.2:** Discrete wavelet families implemented in `PyWavelets`.

Wavelets are localized in both time and frequency, giving them an advantage in signal processing applications that require finer granularity in time, over the Fourier transform which is only localized in frequency. A wavelet analysis can identify changes in frequency (e.g. information) that are inherently localized in the time domain. The wavelet transform analyses a signal in multiple passes at varying scales, which means that the size of the portion of signal under analysis varies as opposed to the Windowed Fourier Transform in which the size of window remains the same. This leads to a better resolution in both time and frequency spaces with wavelet analysis as opposed to the Windowed Fourier Transform. The wavelet transform is used as an alternate method to Windowed Fourier transform [58] when a varying scaling window is needed. Wavelet analysis is a powerful tool to analyse power variations in a time series signal [59]. Power variations in a time series signal are studied to notice patterns, statistics and the evolution of a signal that can provide insight about the data.

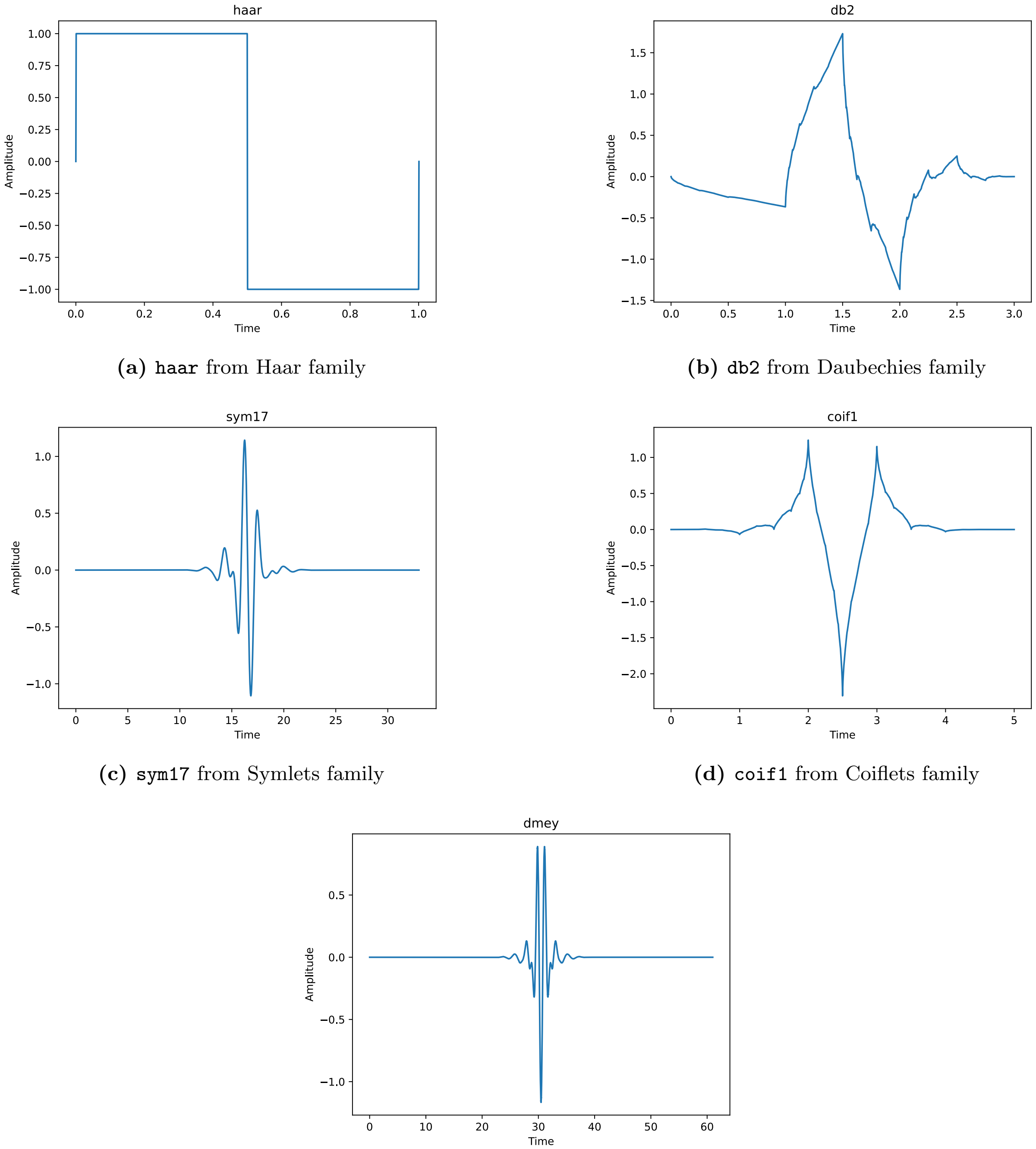
Wavelet analysis was used to remove noise from the nanopore signal in this project, referred to as wavelet denoising from here on. The wavelet denoising procedure has three steps:

1. Decomposition of signal: Decomposition of signal is started by choosing a wavelet and a level of decomposition `N`. A level of decomposition refers to the maximum level up to which a signal can be decomposed. It is a positive integer less than or equal to $log_2 L$, where `L` is the length of a signal. Varying decomposition levels offers varying granularity of the composition and information in the signal.

Deconstructing a signal provides us with approximation coefficients and detail coefficients, which can be used later to reconstruct the signal.

2. Filtering noise from signal: For each level `1 - N`, a threshold value is selected and soft thresholding is applied to the detail coefficients. Soft thresholding [60] refers to shrinking the coefficients of a deconstructed signal towards zero to remove background noise. Filtering the coefficients under a certain threshold is used to eliminate small details from a signal which are often associated with noise, making the signal curve smoother.

3. Reconstruction of signal: The signal is finally reconstructed based on the original approximation coefficients and the modified detail coefficients.



**(a)** `haar` from Haar family

**(b)** `db2` from Daubechies family

**(c)** `sym17` from Symlets family

**(d)** `coif1` from Coiflets family

**(e)** `dmey` from Discrete approximation of Meyer family

**Figure 2.3:** Wavelet examples from `PyWavelets`

# 3 Literature Review

The pioneering DNA sequencing methods were invented by Maxam and Gilbert [4], and Sanger [5] in 1977 which started the first generation sequencing era. The Sanger method also known as chain-termination or dideoxy method was the popular choice until the late 20th century. Developments in biochemistry such as the invention of the Polymerase Chain Reaction (PCR) [16] in 1984 kept the Sanger sequencing method relevant into the 21st century and the Human Genome Project [7] phase. A major drawback of first generation sequencing methods was that Sanger and other methods sequenced short reads consisting of 300-1000 base pairs. Researchers sequenced short strands of DNA with this method and later stitched them together computationally to create longer sequences. In the stitching process, shorter strands were connected using common overlapping regions between multiple strands which contributed to incorrect sequencing of repetitive regions. A large amount of sample DNA is required by the first generation methods for sequencing which is usually achieved by sample amplification. Sample amplification methods can introduce significant errors in the sequencing process leading to inaccurate sequencing. Another drawback of the first generation was the high cost associated with sequencing. The cost of sequencing a complete human genome in the Human Genome Project was USD 3 billion over ten years. In 2007 there was a push from the US National Human Genome Research Institute (NHGRI) to reduce the cost of genome sequencing to USD1000. Concurrent to the Human Genome Project, second generation sequencing methods started being developed during the mid-20th century that were vastly more scalable and parallel than first generation sequencing methods. These methods reduced the cost of sequencing a human genome from millions of dollars to USD1000. Among the most popular second generation sequencing technologies are Roche 454 sequencing, IonTorrent sequencing, Illumina sequencing and ABI/SOLiD sequencing [61]. Illumina is one of the most widely used second generation sequencing technologies used commercially and academically. In the recent years, third generation sequencing technologies have emerged that sequence DNA in real-time on a single molecule level, producing substantially longer reads [7]. Currently the most widely used third generation sequencing technologies are PacBio SMRT sequencing [8] by Pacific Biosciences and nanopore sequencing [9] by Oxford Nanopore Technologies.

All existing sequencing technologies come with their own challenges and strengths [62]. High costs are associated with instruments, sample preparation and amplification, and researcher's time in first and second generation sequencing technologies. Short read lengths in first and second generation sequencing lead to high overhead in post-experiment analysis. Despite of the higher cost and expertise required to handle the experimental apparatus, accurate DNA sequencing by first and second generation methods has transformed genetic research.

Nanopore sequencing is a third generation sequencing technology that is lower cost and offers long read sequencing and real time analysis [63]. The lower accuracy of nanopore sequencing, even with the many improvements made in hardware and software components over the years, still remains a challenge. The high-throughput and long-read properties of third generation sequencing technologies have helped make progress in genome sequencing and research over the last two decades and have opened doors for accessible and rapid genome sequencing and assembly [63]. Nanopore sequencing offers great potential for real-time genome sequencing. The ability of nanopore sequencing to produce long reads makes the genome assembly much faster regardless of the size of the genome, making it an ideal sequencing method for small organisms like *Escherichia coli* K-12 MG16 [31] or much larger organisms including humans [64]. However, the accuracy of these generated sequences is still a limitation which can hinder widespread reliance on commercial `ONT` devices [65]. Factors that can contribute to the inaccuracy of nanopore sequencing include type and choice of the protein nanopore, control enzyme used to control the DNA translocation speed, measurement of electric signal, and basecalling methods. Efforts have been made to improve the accuracy of nanopore sequencing devices by improving the nanopore proteins [27] and controlling DNA translocation speed through the pore [28]. The process of basecalling has been improved [29] by employing machine learning techniques and improving the data processing pipelines [66]. Due to these updates in chemistry and software tools, the accuracy of nanopore sequencing has made improvement from $< 60\%$ to approximately 85% compared to reference genomes [26]. Among the most popular choices of machine learning algorithms for use in basecallers are Hidden Markov Models (HMMs) [29] and Recurrent Neural Networks (RNNs) [30]. RNN-based basecallers have proven to be the more popular choice as RNNs are not reliant on sequence length or sequence repetition [67], particularly promising for plant genomes that are known to contain repetitive sequences [68]. There are methods now available for post-processing basecalls provided by tools like `Nanocorrect` and `Nanopolish` to produce higher quality reads [69] and optimizing the genome assembly provided by tools like `Canu`.

In this thesis, the focus was on reducing the raw nanopore signal noise before basecalling and analysis. To remove the noise from a nanopore signal, it is necessary to first understand the origin and characteristics of said noise [44]. Noise characteristics vary depending on the type of protein nanopores and their surface properties [45]. Based on these properties the noise can be reduced by modifying the voltage, capacitance and current across the nanopore. However, these hardware factors are beyond the scope of this thesis. We seek to improve the final signal that a nanopore device produces as output. We hypothesize that wavelet-based filtering can reduce noise in nanopore signals significantly, leading to better quality and high SNR of the nanopore signal because of existing studies that demonstrated that noise in nanopore signal effects the error rate [23] and wavelet transform can be used to improve SNR in an ECG signal [70]. Denoised signals can be used for training basecalling models. The inspiration to use wavelet analysis was drawn from existing studies that have utilized wavelet analysis to achieve better SNR in their applications. Wavelet analysis has applications in various domains and its effect is explained in a few examples:

- Fault identification system for satellites [71]: Wavelet analysis was used to measure SNR and extract patterns from unattended land terminals for satellite communications. These patterns were matched with existing known fault signature patterns and the fault was identified. The study found a success ratio of 80% fault identification using wavelet techniques.

- Cardiac activity system [72]: Electrocardiogram signals (ECS) are electrical excitation patterns of the nervous system in human heart. In this short study, the authors suggested the use of wavelet analysis on electrocardiogram signals to detect cardiovascular inconsistencies.
- Ion channel SNR increase [73]: Wavelet denoising has proven to be more efficient in increasing the SNR for a nanopore and ion channel signal than the traditional Bessel filtering method. The study was conducted on nanopores that analog ion channel proteins in live cells that regulate cell functions such as muscle contraction, hormone release and the life-cycle of cells.

Evaluating the results of wavelet analysis in these studies provides us with confidence that wavelet analysis is a reasonable choice for reducing noise in signals from various applications and origins. This thesis used the existing wavelet analysis framework provided by `PyWavelets` [74]. Well-suited parameter values for the level of decomposition and threshold were picked, while keeping all other parameters and setup constant. Parameter values and choice of wavelet for denoising varies across different applications and studies in existing literature were considered as the strating point to determine the parameter space. A study of wavelet transform parameters for 3d surface filtering showed levels 9 and 8 as top performers for different magnification levels [75]. Similarly a study conducted on classification of multispectral images of forest vegetation determined that the `haar` wavelet performed best at the 3rd level of decomposition while levels 4 through 10 gave similar results [76]. The best-suited parameters for this thesis work are described in Chapter 5. This thesis does not explore the mechanisms of `PyWavelets`, wavelet transform or the wavelets themselves.

# 4 Experiment and Methodology

This thesis explores the use of wavelet analysis for removing noise from raw nanopore signals and creating a custom model for basecalling using the denoised signal. Three datasets were used for the experiments as explained in section 4.1. The discrete wavelet families provided by `PyWavelets` were tried with varying combinations of level of decomposition and threshold and the best performing wavelet and parameter combination that worked for the datasets under consideration were picked.

Each dataset was randomly divided into two subsets: a training dataset and a testing dataset. The training and testing datasets were kept completely separate from the start to the end of the experiment. Figure 4.1 depicts the workflow for this project.

The experiment pipeline consisted of five main steps:

1. Random division of each dataset into training and testing datasets. The training and testing subsets were kept approximately equal in size and number of files.

2. Reduction of noise in the training and testing datasets to create denoised versions of each dataset. The process of denoising is explained in Section 4.3.

3. Creation of basecalling models using raw and denoised training datasets. The models created using the raw training dataset were used as a control to establish a baseline of basecalling statistics for each dataset. `Taiyaki` toolkit by ONT was used to generate custom models for the experiments. Detailed process of model generation is described in Section 4.3.1.

4. Basecall of testing datasets with our newly generated custom models.

5. Analysis of the basecalled sequence and comparison of results.

Custom models were created for this thesis as depicted in the experiment pipeline in Figure 4.1. The following models have been used for this project:

1. Default `Guppy` model: This is the model provided by the `Guppy` basecaller.

2. Species specific model: This model was created using `Taiyaki` toolkit by `ONT` with raw signal for *Bacteriophage lambda* training dataset.

3. Denoised models: These models were created using `Taiyaki` toolkit by `ONT` with denoised signal for *Bacteriophage lambda* training datasets. Multiple models were created for varying denoising parameters.

4. Multi-species model: Multi-species models were created using nanopore data from multiple organisms.

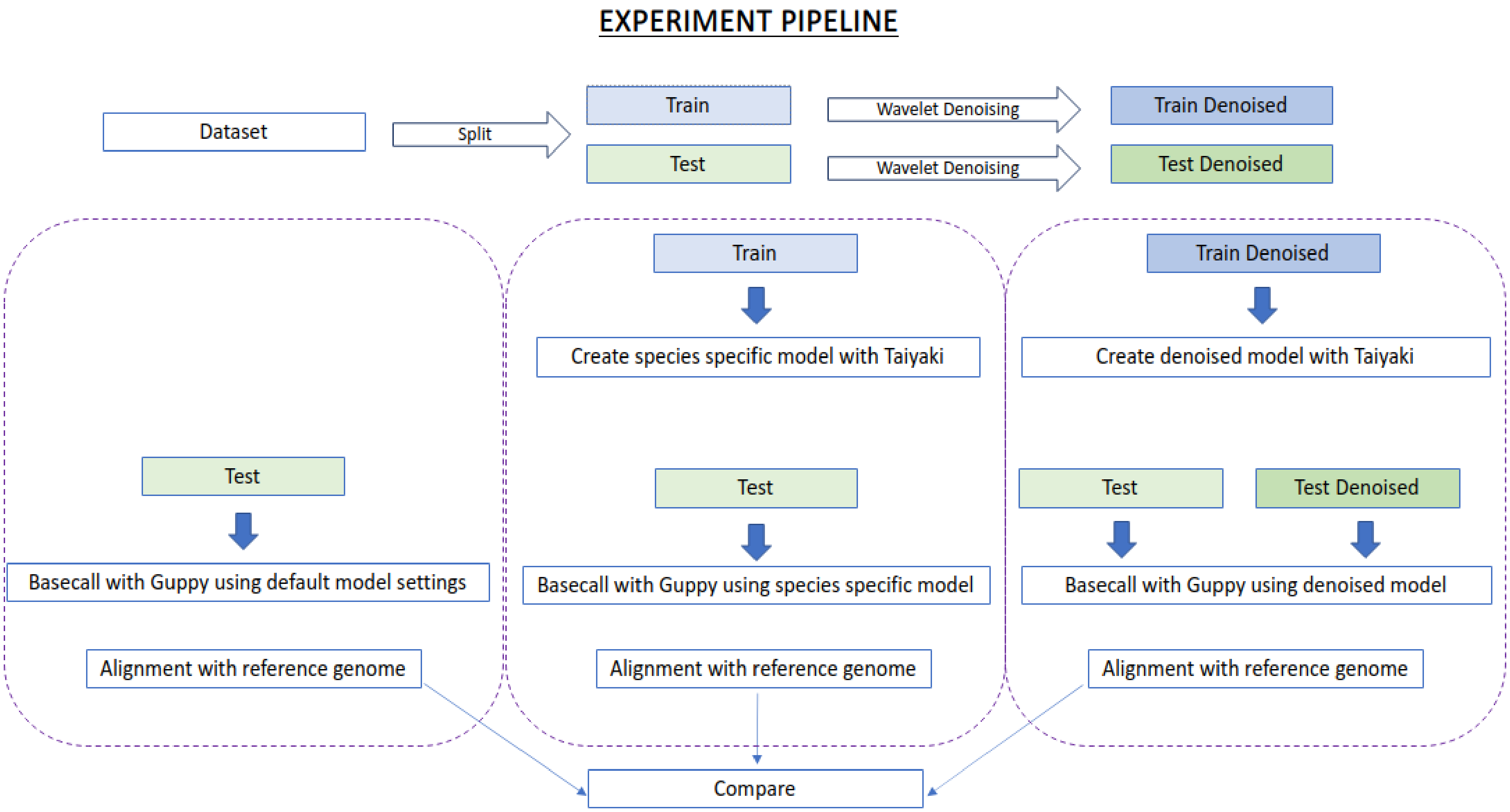

EXPERIMENT PIPELINE
Dataset
Split
Train
Test
Wavelet Denoising
Wavelet Denoising
Train Denoised
Test Denoised
Test
Basecall with Guppy using default model settings
Alignment with reference genome
Train
Create species specific model with Taiyaki
Test
Basecall with Guppy using species specific model
Alignment with reference genome
Train Denoised
Create denoised model with Taiyaki
Test
Test Denoised
Basecall with Guppy using denoised model
Alignment with reference genome
Compare

**Figure 4.1:** Experiment pipeline

## 4.1 Data

The data used in this project was generated by collaborators and was provided for the bioinformatics experiments described in this thesis. Table 4.1 describes the three different primary datasets used in this thesis. All data files used for this project were in `fast5` format which is based on `hdf5` format.

The *Bacteriophage lambda* reference genome consists of 48,502 base pairs. The *Bacteriophage lambda* genome is classically used as the first step in bioinformatics pipeline creation due to its simplicity and small size [77]. It was the first genome to be fully sequenced, and continues to be used as a benchmark dataset. *Bacteriophage lambda* was used as the primary dataset in this thesis. The training dataset has a 4117 times coverage of the *Bacteriophage lambda* reference genome, and the testing dataset has a 4944 times coverage of the reference genome.

The experiment was also run for a *Bos taurus* dataset provided by collaborators (from bulls and calves sired by the bulls in the collection) [78]. The reference genome of *Bos taurus* conists of 2.7 gigabases, considerably larger than the *Bacteriophage lambda* reference genome. The purpose of running the same experiment on two organisms of a drastically different genome size is to see the effect of wavelet filtering on varying sizes of genomes and datasets.

The `Taiyaki` toolkit provides a training walk-through, referred to as `Taiyaki Walk-through` in this document, with an example dataset to create a `Guppy` compatible basecalling model. In addition to *Bacteriophage lambda* and *Bos taurus*, reads provided by `Taiyaki Walk-through` for yeast, *E. coli* and human genomes to create custom models were used to observe the effect wavelets have on models that are created using completely different groups of species than the testing dataset.

Table 4.1 summarises the properties of our training and testing datasets. Mean percentage identity, mean read length, mean read quality, and total bases were obtained by using `NanoStat` 1.4.0 from the NanoPack package [35].

| **Properties** | ***Bacteriophage lambda*** | | **Taiyaki Walk-through** | | **Cattle** | |
|---|---|---|---|---|---|---|
| | **Training** | **Testing** | **Training** | **Testing** | **Training** | **Testing** |
| **Organisms** | *Enterobacteria phage lambda* (NC_001416.1[1]) | | *Escherichia coli* (SCS110[2]),<br>*H. sapiens* (NA12878[3]),<br>*S. cerevisiae* (NCYC1052[4]) | | *Bos taurus* (ARS-UCD1.2[5]) | |
| **DNA Source and Preparation** | Source DNA was prepared as per the manufacturer's burn-in experiment for the `MinION` platform and included an equimolar concentration of *E. coli* K12 substr. MG1655 and *Bacteriophage lambda*. | | Provided by `ONT` without description. | | DNA was isolated from blood draws as described previously [78]. Purified DNA was prepared for sequencing using the Rapid Barcoding Sequencing kit (SQK-RBK004, from Oxford Nanopore). | |
| **Data Generation** | Data was generated using a FLO_MIN104 flow cell. MinKNOW version 0.51.1.62 was used to control the sequencing run as per the manufacturers recommendations. | | No description provided by `ONT`. | | Sequencing was conducted on FLO-MIN 106D flow cells using MinKNOW as per the manufacturer's suggestions. | |
| **Reference Genome Size** | 48,502 (48.5 Kilobases) | | 6,193,545,673 (6.2 Gigabases) (all organisms combined) | | 2,715,853,792 (2.7 Gigabases) | |
| **Total Bases** | 199,675,899 (199.7 Megabases) | 239,807,324 (239.8 Megabases) | 917,176 (917.2 Kilobases) | – | 2,856,099,801 (2.9 Gigabases) | 2,862,591,360 (2.9 Gigabases) |

[1] NCBI reference genome

[2] *E. Coli strain reference*

[3] NCBI reference genome

[4] National Collection of Yeast Cultures strain reference

[5] NCBI reference genome

| | | | | | | |
|---|---|---|---|---|---|---|
| **Coverage w.r.t Reference Genome** | 4116.9 | 4944.3 | 0.00015 | – | 1.05 | 1.05 |
| **Size on Disk** | 20GB | 22GB | 4.5GB | – | 69GB | 69GB |
| **Mean Read Length** | 6,523.4 (6.5 Kilobases) | 6,126.4 (6.1 Kilobases) | 3,902.9 (3.9 Kilobases) | – | 7,155.4 (7.2 Kilobases) | 7,185.1 (7.2 Kilobases) |
| **Mean Read Quality** | 9.2 | 11.0 | 15.4 | – | 10.7 | 10.7 |
| **Mean Percentage Identity w.r.t Reference Genome** | 86.2 | 90.6 | 89.9 | – | 88.1 | 88.2 |

**Table 4.1:** Datasets' properties.

## 4.2 Basecalling and Alignment

`Guppy` was used as the basecaller in this project. `Guppy` is an RNN-based basecaller by `ONT` that includes post-processing features. It allows customization of configuration for data analysis and comes integrated with some nanopore sequencing devices such as `MinKNOW`. Initially, experiments were conducted using `Scrappie` [30], `Albacore` [30], but `Guppy` was chosen due to it's ease of use and ability to integrate user models. All tests were run with the CPU version of `Guppy` 4.2.3 on Ubuntu 18.04.5. The configuration for `Guppy` was kept constant throughout the process using the configuration file `dna_r9.4.1_450bps_hac.cfg` provided in the Guppy installation. The model file `template_r9.4.1_450bps_hac.json` as provided by the `Guppy` package was used for basecalls in the control groups. We used `minimap2` for individual alignments, and for collective model comparison `NanoPlot` (v1.20.0) and `NanoStat` (v1.4.0) from the `NanoPack` package [35] were used.

## 4.3 Denoising

Implementation of wavelet analysis provided by `PyWavelets` [74] was used for the wavelet denoising process. Sample code for denoising nanopore data is given in Appendix A. `PyWavelets` provides implementations for 14 common wavelet families and their member wavelets. Table 2.2 provides brief overview about each wavelet family considered for this thesis. These wavelets are implemented in `PyWavelets`. Wavelets from each of these families have unique scaling and wavelet functions. These functions and their shapes determine which wavelet will be the best choice for various applications. For example, wavelets Daubechies-3 and Symlets-3 have been proven the best choice for removing noise from ECG signals [70]. Due to the large number of candidates it is important to narrow the list of potential wavelets to a practical and testable set. A shortlisting strategy based on `MapQ` score by `minimap2` was used to determine the best choice of wavelet for removing noise from nanopore sequencing data.

Digital signal processing techniques were used as the inspiration for shortlisting the parameter space and to estimated SNR improvement after denoising our data. The implementation of Welch's method [79] method provided by `SciPy` [80] was used to determine the signal band and the noise band. The signal was found to reside almost entirely in the lower 1/4 of the frequency range. Signal power is defined simply as the largest spectral peak, and the noise floor as the average power in the upper 1/4 of the frequency range in this thesis. Keeping with convention the power levels are calculated in decibels (dB), and were created using the picoampere primary measurement and assuming a resistance of 1 ohm.

$$\text{dB} = 10 \times \log_{10}(\text{picoamphere}^2)$$

Using the Welch method, noise floor values were first extracted from sample signals. The noise floor was then subtracted from the highest power peak of the signal to calculate SNR.

$$\text{SNR} = \text{signal} - \text{noise}$$

| Property | Original Signal | Denoised Signal |
|---|---|---|
| Max signal power | -213.2383 dB | -213.2434 dB |
| Noise in signal | -229.4417 dB | -230.3154 dB |
| SNR from FFT | 16.2034 dB | 17.0719 |
| SNR improvement | 0.8685 dB | |
| Signal change | -0.0052 dB | |
| Noise floor change | -0.8737 dB | |

**Table 4.2:** Change in signal after denoising signal with the `haar` wavelet at level of decomposition 4 and threshold 0.04

Figure 4.2a and 4.2b show an example comparison of the frequency spectra between the original and denoised signal. The highlighted portion between 1500 and 2000 on the frequency axis represents the upper 1/4 range of the frequency used to determine the noise in signal.

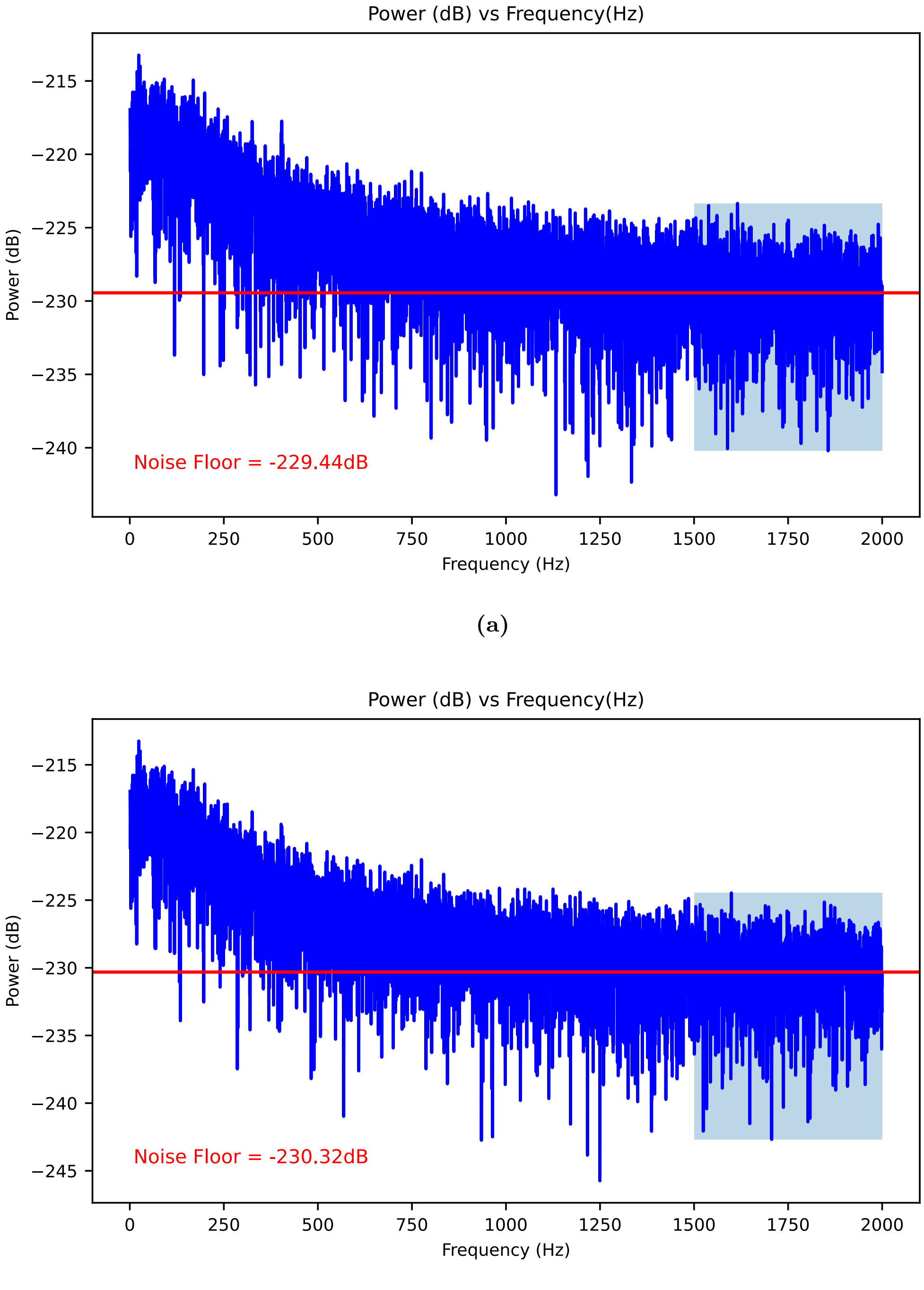
Any filtering of the original signal may distort the signal or even add noise (in the case of over filtering). The goal is to minimize the signal loss, while maximizing the noise reduction. Table 4.2 details the same example as Figure 4.2 which has a very small change in signal and a significant reduction in noise. Table 4.2 shows the increase in SNR by 0.8685dB or 5.4% compared to the original nanopore signal during the parameter shortlisting stage of the experimental process. The change in signal is small $-0.0052$dB while the change in noise is $-0.8737$dB. This provides an example of losing significantly more noise from the signal and keeping the signal detail intact.

All wavelets that only offer continuous wavelet transforms were removed from consideration, because our data is discrete, leaving seven wavelet families. Those seven families were tested using a subset of the *Bacteriophage lambda* dataset. Comparison of `MapQ` scores generated by `minimap2` was used as an initial estimate. After settling on a single wavelet based on the observations described in Chapter 5, the possibilities for level of decomposition and threshold were narrowed down. Varying the parameters of wavelet transform affects the results [81], inclining us to keep a few options for both level of decomposition and threshold to observe the effect on basecalling.

In addition to denoising the training dataset, denoised copies of the testing dataset were created for each shortlisted combination to compare any differences in result during testing. It was our hypothesis that best results will be obtained using denoised datasets for both training basecall models and testing.

### 4.3.1 Creating the Model

Using the `Taiyaki v(5.0.0)` toolset new models for `Guppy` were created for basecalling. `Taiyaki` is a tool developed by `ONT` to train models for basecalling. For this project `Taiyaki` was used to create custom models compatible with `Guppy`. The models generated by `Taiyaki` were used to basecall different versions of the test dataset and the performance of these basecalls was compared with the control configuration of each dataset.



**(a)**

**(b)**

**Figure 4.2:** Comparison of noise floor in raw signal and denoised signal. The shaded portion between 1500 and 2000 Hz on the x-axis represents the region used to calculate the noise floor. SNR in a sample nanopore signal is 16.20 dB. After denoising a sample nanopore signal with `haar` wavelet, level of decomposition = 4 and threshold = 0.04, SNR is 17.07 dB

# 5 Results

This thesis work determines the most suitable wavelet for pre-filtering nanopore signal, best performing denoising parameters and the overall impact they had on basecalling. A significant improvement in basecalling accuracy was achieved by using custom models generated with denoised nanopore data.

## 5.1 Wavelet Analysis

In order to generate the models, first a single wavelet and accompanying parameters for wavelet denoising were chosen. Figure 5.1 was generated by basecalling denoised test dataset with varying combinations of wavelets, decomposition level and threshold values for all the discrete wavelet families. There are 7 wavelet families and 9 values each for decomposition level (0-9) and threshold (0.01-0.09) that were considered. For each wavelet family, Figure 5.1 presents the `MapQ` score for all 81 combinations of decomposition level and threshold. For all combinations of decomposition level and threshold, `MapQ` scores were analysed for each wavelet family. The default model displays the most consistently high `MapQ` score for raw test signal. This was expected as the model has been trained for raw test signal. The goal of this comparison was to detect the wavelet family that affects the basecalling process the least and maintains a high `MapQ` score.

The Haar wavelet family stands out amongst all wavelet families in terms of the highest mean `MapQ` score. This edge of the `haar` wavelet over others in terms of `MapQ` suggests it should be preferred over the other wavelets for this use case. The shape of the `haar` wavelet also resembles the near step-wise shape of the nanopore signal, and it is our hypothesis that this property of the `haar` wavelet will be useful in denoising the nanopore signal.

The effect of `haar` wavelet from the Haar wavelet family was then observed by only considering `haar` wavelet during denoising the subset of testing dataset. Figure 5.2 shows the impact of the `haar` wavelet transform on the average `MapQ` score distribution for each level and threshold combination for the *Bacteriophage lambda* dataset. Sample code for denoising nanopore data is given in Appendix A.

The peaks can be observed at mid-range values for both level of decomposition and threshold, suggesting a high chance that these parameters will perform better for our use case. A dip of `MapQ` score at higher extremes for threshold values suggests that these parameters are not ideal for denoising, and they diminish the quality of the signal and basecalls produced. It was further noticed that most combinations of moderate threshold and decomposition level values maintain a stable `MapQ` score for the *Bacteriophage lambda* dataset. This stability in the MapQ score is a good indicator that the experiment was headed in the right direction

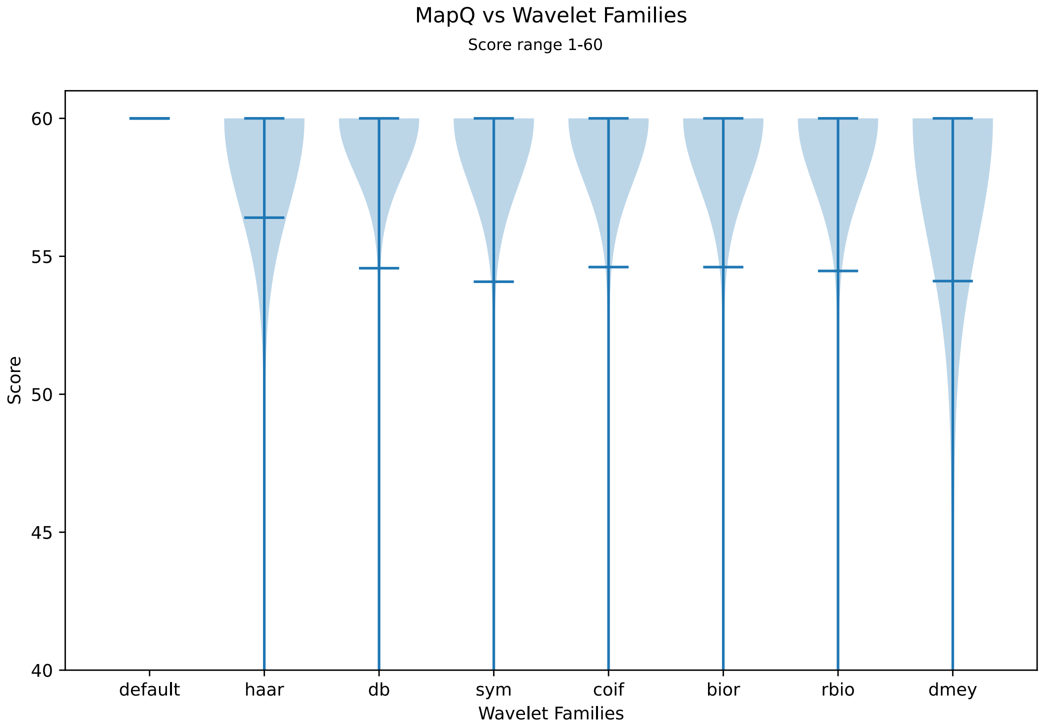


**Figure 5.1:** `MapQ` score distribution across all wavelets in each wavelet familiy

while narrowing the parameter space.

Wavelet filtering should be applied carefully, using appropriate denoising parameters for any given use case. As Figure 5.2 indicates, incorrect use of wavelet filtering on nanopore signal can be detrimental to an experiment. In this thesis, the method and result for picking out the best parameter combinations for denoising the nanopore signal is presented. Similar process can be used to determine the optimal settings for any other experiment, inside and outside the realm of nanopore sequencing.

As Figure 5.1 shows a greater improvement in the average `MapQ` score when using the Haar wavelet family compared to other wavelet families, and Figure 5.2 provides an insight to the `MapQ` distribution across different combinations of decomposition level and threshold, giving us a starting point to consider shortlisting the values of those parameters as well. As the Haar wavelet family consistently gave us the best decomposition and reconstruction of our signal, the single wavelet `haar` from the Haar family was chosen as the final candidate for this thesis. We believe that the step-wise shape, which resembles the expected output from a nanopore signal [82], is the reason for its superior performance in denoising nanopore signal data compared to other wavelets.

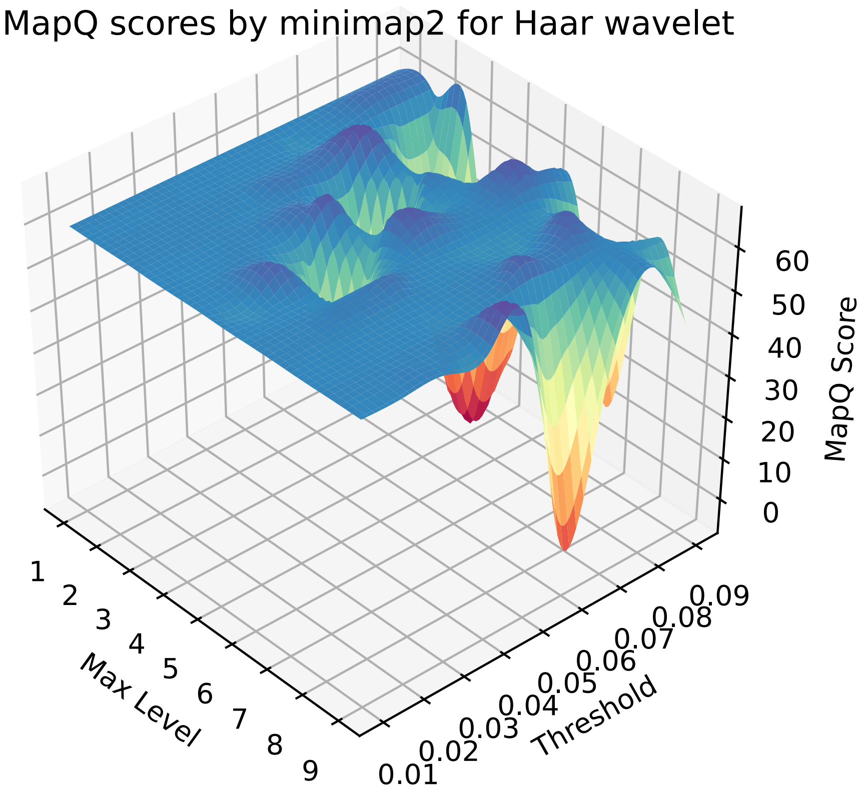


**Figure 5.2:** `MapQ` score distribution for `haar` wavelet across all threshold and level combinations

| **Decomposition** | **Threshold** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Level** | **0.01** | **0.02** | **0.03** | **0.04** | **0.05** | **0.06** | **0.07** | **0.08** | **0.09** |
| **1** | 0.25 | 0.41 | 0.51 | 0.58 | 0.62 | 0.64 | 0.65 | 0.66 | 0.66 |
| **2** | 0.26 | 0.42 | 0.53 | 0.61 | 0.67 | 0.71 | 0.73 | 0.76 | 0.78 |
| **3** | 0.26 | 0.43 | 0.56 | 0.65 | 0.72 | 0.78 | 0.82 | 0.86 | 0.90 |
| **4** | 0.26 | 0.43 | 0.57 | 0.67 | 0.75 | 0.82 | 0.88 | 0.93 | 0.99 |
| **5** | 0.25 | 0.42 | 0.55 | 0.65 | 0.73 | 0.80 | 0.87 | 0.93 | 0.99 |
| **6** | 0.23 | 0.39 | 0.5 | 0.6 | 0.67 | 0.74 | 0.8 | 0.86 | 0.92 |
| **7** | 0.22 | 0.36 | 0.47 | 0.54 | 0.61 | 0.66 | 0.72 | 0.77 | 0.82 |
| **8** | 0.21 | 0.34 | 0.44 | 0.51 | 0.56 | 0.6 | 0.64 | 0.68 | 0.72 |
| **9** | 0.21 | 0.33 | 0.42 | 0.48 | 0.52 | 0.56 | 0.59 | 0.61 | 0.64 |

**Table 5.1:** Mean SNR improvement for each level-threshold combination

## 5.2 Denoising Parameters

For the *Bacteriophage lambda* dataset, all combinations of thresholds (0.01 to 0.09) and levels of decomposition (1 to 9) were tested for their estimated SNR improvement. The signal to noise ratio was calculated first for a subset of the original signals and then for the denoised signals with all combinations of levels of decomposition and thresholds. For each threshold and level of decomposition combination the mean of change in SNR was considered to observe the pattern of change. Figure 5.3 shows the plotted mean SNR improvements from Table 5.1. To choose the optimal level of decomposition and threshold, the extent of smoothing of signal that is acceptable to avoid over-filtering was considered. It is evident from Figure 5.3 that the optimal range of levels is between 3 and 5 with an obvious inflection at a threshold between 0.03 and 0.05 range. Threshold values over 0.05 signify the start of over-filtering because we start losing significant signal detail along with the noise. Any threshold under 0.03 does not produce a significant enough improvement in SNR to investigate. Loss of signal power is shown by a flattening of the curve at higher thresholds. This leaves the mid-range values that we can shortlist further by comparing consecutive values to determine if there is any significant improvement between them. Figure 5.4 shows the mean improvement in SNR in a 3d plane, for Table 5.1. This 3d plot further emphasizes the improvement in SNR for mid-range decomposition level and threshold values. The parameter values of interest correspond to the green region in the 3d plane.

A representation for median SNR improvement is provided in the Appendix C in Table C.1, and Figures C.1 and C.2. The change in signal and change in noise were both considered independently as depicted in Table 4.2. The mean SNR improvement in Table 5.1 was also taken into account to narrow down the parameter space. A few extreme values were picked for these parameters to serve as an experimental control. Table 5.2 shows the list of chosen parameter values that will be used for this experiment.

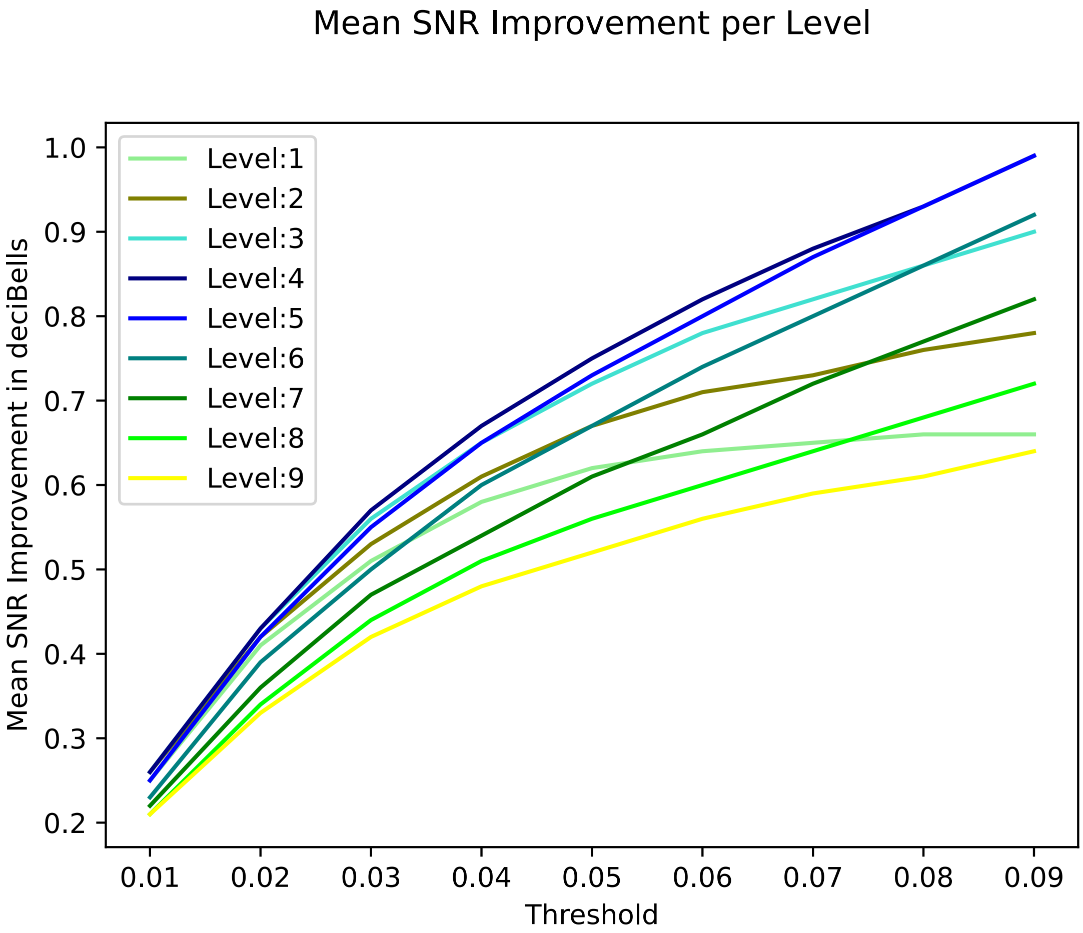


**Figure 5.3:** Mean SNR improvement for each level-threshold combination

| Parameter | Shortlisted based on SNR improvement |
|---|---|
| Wavelet | `haar` |
| Level of decomposition | 1 [1] , 3, 4, 5 |
| Threshold | 0.01[2], 0.03, 0.04, 0.06, 0.09[3] |

**Table 5.2:** Parameters shortlisted after SNR analysis.

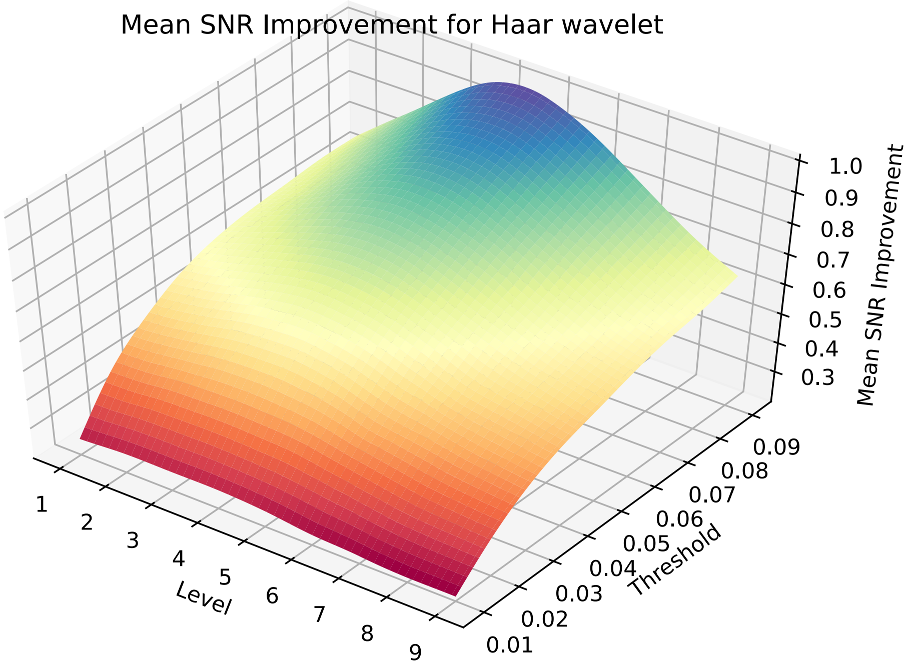


**Figure 5.4:** Mean SNR improvement for each level-threshold combination

## 5.3 Basecalling with Custom Models

Now with the parameter space reduced, the experiments were performed as shown in Figure 4.1. Denoised versions of the training and testing datasets were created, custom models were then generated using the various versions of training datasets and all versions of testing datasets were basecalled. Mean percentage identity and mean read quality generated by `NanoStat` were used as the measure for comparing the accuracy of custom models against default `Guppy` model. Note that the read quality calculated by `NanoStat` is different from the `MapQ` score. Basecalling the *Bacteriophage lambda* dataset using denoised signal improved the mean percentage identity of basecalls by 5.3%. A complete list of experiment runs and basecall statistics is shown in Table B.1. The best performing models from Table B.1 have been presented in Figure 5.5 and Table 5.3. Consider Figure 5.5 to observe the number of matches, mismatches, insertions and deletions in the basecalls produced by `Guppy`. The experiments were divided into three categories: control group, raw testing datasets and denoised testing datasets. The control group will serve as the benchmark to be compared against for all other custom models.

Figure 5.5a shows the matches, mismatches, insertions and deletions in the basecalls produced by using the control group models, including the default `Guppy` model, *Bacteriophage lambda* species specific model and a generic `Taiyaki Walk-through` generic model. Within the control group the best performer is the *Bacteriophage lambda* species specific model with the most number of matches, followed by the default `Guppy` model. The `Taiyaki Walk-through` generic species model performs the worst, as can be expected from the composition of its training dataset. The training dataset was not diverse and only contained reads for Yeast, Ecoli and Human.

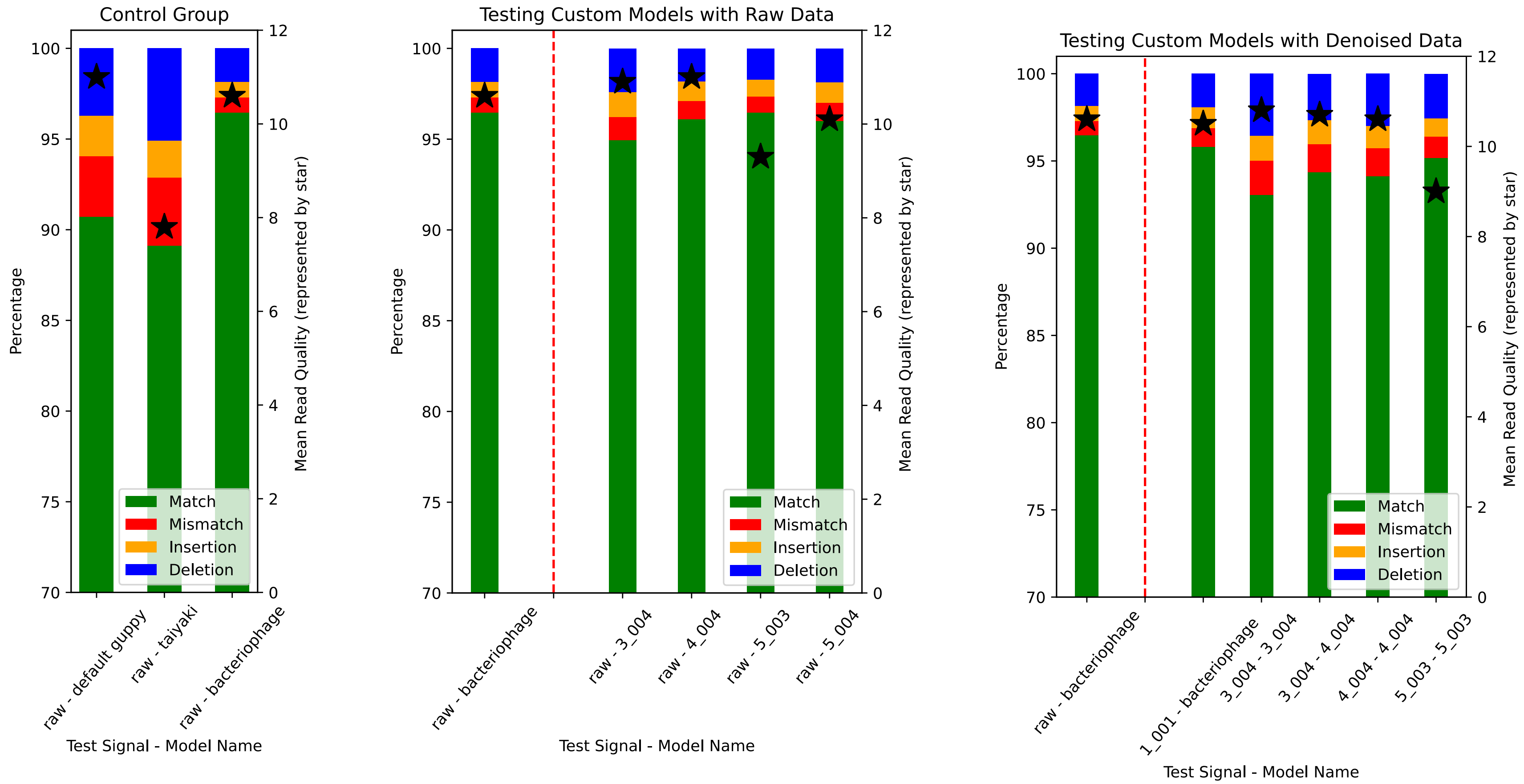
The custom denoised models were then tested with raw signal as well as with denoised testing datasets. Figure 5.5b shows the performance of our custom generated denoised models against un-processed testing dataset.

Figure 5.5c shows the performance of our models when tested with denoised testing datasets. The unprocessed testing dataset group produced a consistently higher number of matches when compared against the denoised testing datasets group, suggesting that we do not need to denoise the data to be basecalled. As can be observed from Figure 5.5, denoised models decreased the number of insertions, deletions and mismatches - all three indicators being used to evaluate basecalling accuracy and model performance, demonstrating that denoising improves basecalling for all considered failure modes. The results also implied that training for a specific organism is better than training a general pan-organism model.

[1] A low extreme value for level of decomposition

[2] A low extreme value for threshold

[3] A high extreme value for threshold



**(a)** The control group consists of the generic model shipped with `Guppy` basecaller, a model generated with `Taiyaki Walk-through` sample dataset, and a species specific model for *Bacteriophage lambda*.

**(b)** Models created using denoised training datasets were tested with raw testing dataset. The quality score and matches, mismatches, insertions and deletions are then compared against the species specific model. An increase in the number of matches can be seen compared to the control group, as well as an increase in the quality score compared to the species specific model.

**(c)** Models created using denoised training datasets were tested with denoised testing datasets. The quality score and matches, mismatches, insertions and deletions are then compared against the species specific model. An increase in quality and matches can be seen compared to the control group, similar to the raw testing group.

**Figure 5.5:** Comparison of matches, mismatches, insertions and deletions among the top performing models. Stars on each bar represent the Quality Score assigned by `minimap2`

Now consider Table 5.3 as it presents a finer representation of basecalls accuracy and quality. The mean read quality and the mean percentage identity in the control group serve as a benchmark for the performance of custom models. The leading performance of *Bacteriophage lambda* species specific model and low performance of `Taiyaki Walk-through` generic model in the control group were both expected due to the nature of the data. A significant increase in mean percentage identity was noticed while maintaining the mean read quality for both raw and denoised testing groups. Also note that the raw testing group performed better than the denoised testing group. The improvement seen in both the test groups is consistent, providing a check that the improvement did not happen by chance. In addition to the three experiment categories in Figure 5.5, Table 5.3 also enlists a fourth category to present the models created using extreme values for level of decomposition and threshold.

As expected, these models do not perform well compared to the other categories. It is further noticed that the best improvement in both percentage identity and read quality was achieved when the original raw test dataset was basecalled with a denoised model generated with level of $decomposition\ \ level = 4$ and $threshold = 0.04$. The performance of model 4_004 reinforces our initial hypothesis of mid-range parameter values producing high performance basecalls, by denoising the signal effectively. The custom denoised model was slightly behind the *Bacteriophage lambda* species specific model in the percentage identity measure but exceeded it in the mean read quality. It was important to improve the percentage idenitity of basecalls while also improving the reads quality or atleast keeping it consistent to match the generic `Guppy` model's basecalls. A higher percentage identity with a low quality score is not desirable. As expected in our initial hypothesis, the parameter combinations involving the extreme values for decomposition level and threshold did not perform well.

It was observed that the models generated using extreme values for decomposition level and threshold mentioned in Table 5.2 for wavelet denoising performed the worst. This observation proved our initial assumption that higher thresholds and decomposition levels contribute to over-filtering the nanopore signal. Over-filtering leads to loss in signal detail, hence affecting the accuracy of the basecalling process. Our assumption was confirmed that the lower values for both threshold and decomposition level do not filter the noise in the signal to a significant extent, compared to the original raw signal. Such insignificant reduction of noise does not affect the accuracy of the resulting basecalls, making the pre-processing and model generation experiments ineffective. An additional representation of the number of matches alone is shown in Figure C.3.

Figure 5.6 compares the improvement in percentage identity of the newly created model with the denoised training dataset against the default `Guppy` model. Fewer outliers and low-accuracy basecalls can be seen when using custom model compared to basecalls generated by the default `Guppy` configuration. A higher mean percentage idenitity in the custom denoised model was achieved. Figure 5.7 compares the improvement in read quality score of our newly created model with a denoised training dataset against the default `Guppy` model. Similar to Figure 5.6, a significant decrease can be seen in both the number of outliers in our basecalled sequences and higher density in the higher score region. Both percentage identity and read quality become

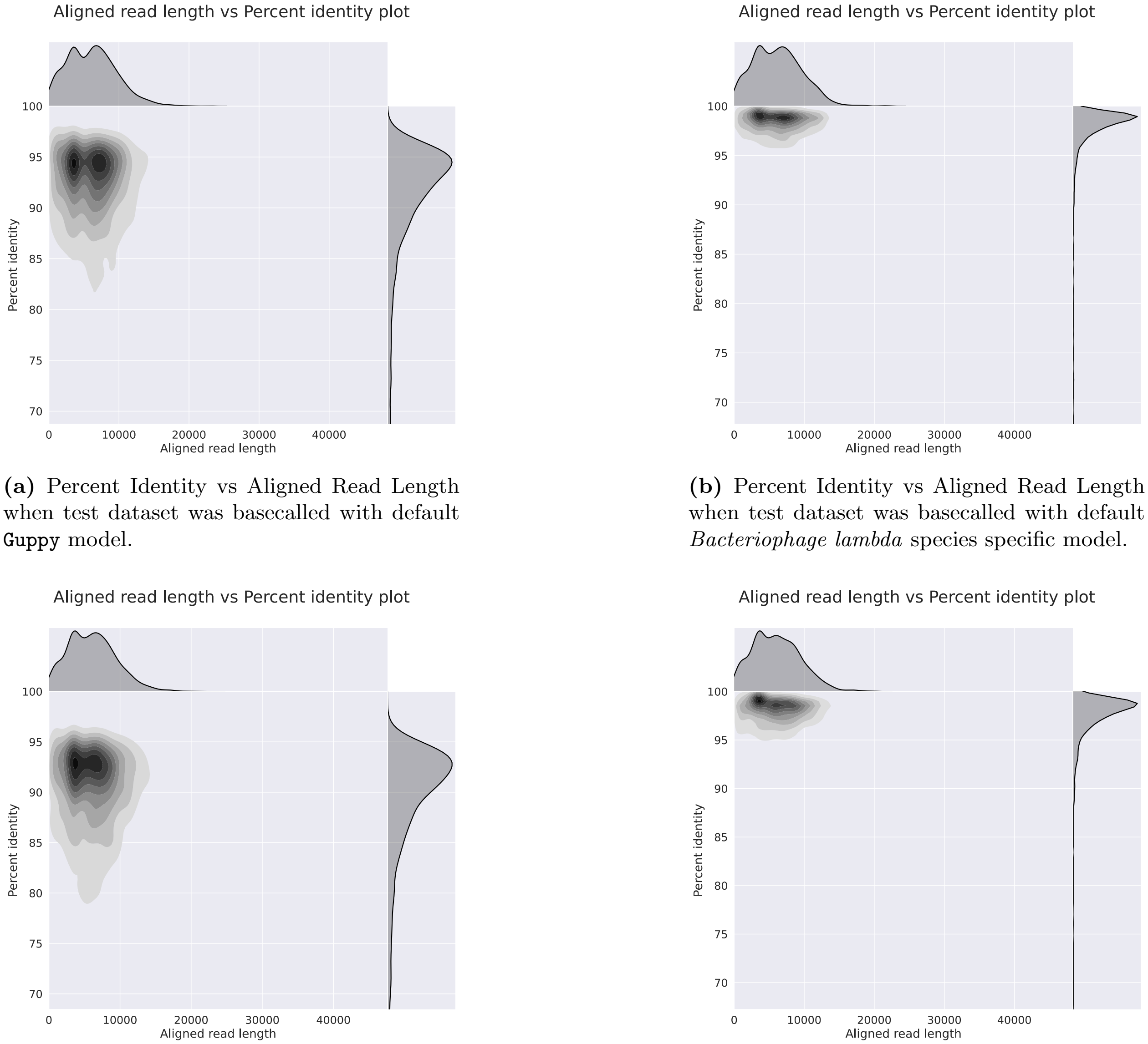
| Test Category | Test dataset | Model | Mean Percentage Identity | Mean Read Quality |
|---|---|---|---|---|
| Control Group | Raw signal | Default `Guppy` model | 90.60 | 11.00 |
| | Raw signal | *Bacteriophage lambda* Model | 96.20 | 10.60 |
| | Raw signal | `Taiyaki walkthrough` generic model | 89.00 | 7.80 |
| Denoised models tested with raw signal | Raw signal | denoised *Bacteriophage lambda* 3_004 | 94.70 | 10.90 |
| | Raw signal | denoised *Bacteriophage lambda* 4_004 | 95.90 | 11.00 |
| | Raw signal | denoised *Bacteriophage lambda* 5_003 | 96.20 | 9.30 |
| | Raw signal | denoised *Bacteriophage lambda* 5_004 | 95.70 | 10.10 |
| Denoised models tested with denoised signal | 1_001 | *Bacteriophage lambda* Model* | 95.60 | 10.50 |
| | 3_004 | denoised *Bacteriophage lambda* 3_004 | 93.40 | 10.80 |
| | 3_004 | denoised *Bacteriophage lambda* 4_004 | 94.60 | 10.70 |
| | 4_004 | denoised *Bacteriophage lambda* 4_004 | 94.40 | 10.60 |
| | 5_003 | denoised *Bacteriophage lambda* 5_003 | 95.20 | 9.00 |
| Models created with extreme parameter values | 1_001 | denoised *Bacteriophage lambda* 1_001 | 95.90 | 10.70 |
| | 3_009 | denoised *Bacteriophage lambda* 3_009 | 91.30 | 8.90 |
| | 4_009 | denoised *Bacteriophage lambda* 4_009 | 86.20 | 9.20 |
| | 5_009 | denoised *Bacteriophage lambda* 5_009 | 86.40 | 9.50 |

**Table 5.3:** Experiment results for control group and top performing models for raw and denoised signals.

a lot more consistent when using custom denoised models, decreasing the variability of basecalled sequences. Less variability in the percentage identity and read quality are indicators of a better basecalling model and overall higher basecalling quality.

Considering all these observations, the wavelet `haar` from the Haar family of wavelets was considered the best choice for removing noise from nanopore signal, along with parameter values $decomposition\ \ level = 4$ and $threshold = 0.04$. Our hypothesis about the extreme values of denoising parameters held true, as they did not perform well compared to the mid-range parameter values.

Models for Cattle, Yeast, Ecoli and Human were then created using the discovered settings. For *Bos taurus* dataset, a decrease in mean percentage identity of 13.4% and a decrease of 0.4 in mean read quality of basecalls was noticed with custom model, as compared to the default model. Yeast, Ecoli and Human genome reads from the `Taiyaki Walk-through` dataset were used to create a standalone model and a hybrid model in combination with *Bacteriophage lambda* reads. In both cases the mean percentage identity was low compared to species specific models. We suspect that the low accuracy of these basecalls is due to much lower coverage of the reference genome by the training datasets. These results can be further investigated in future work.



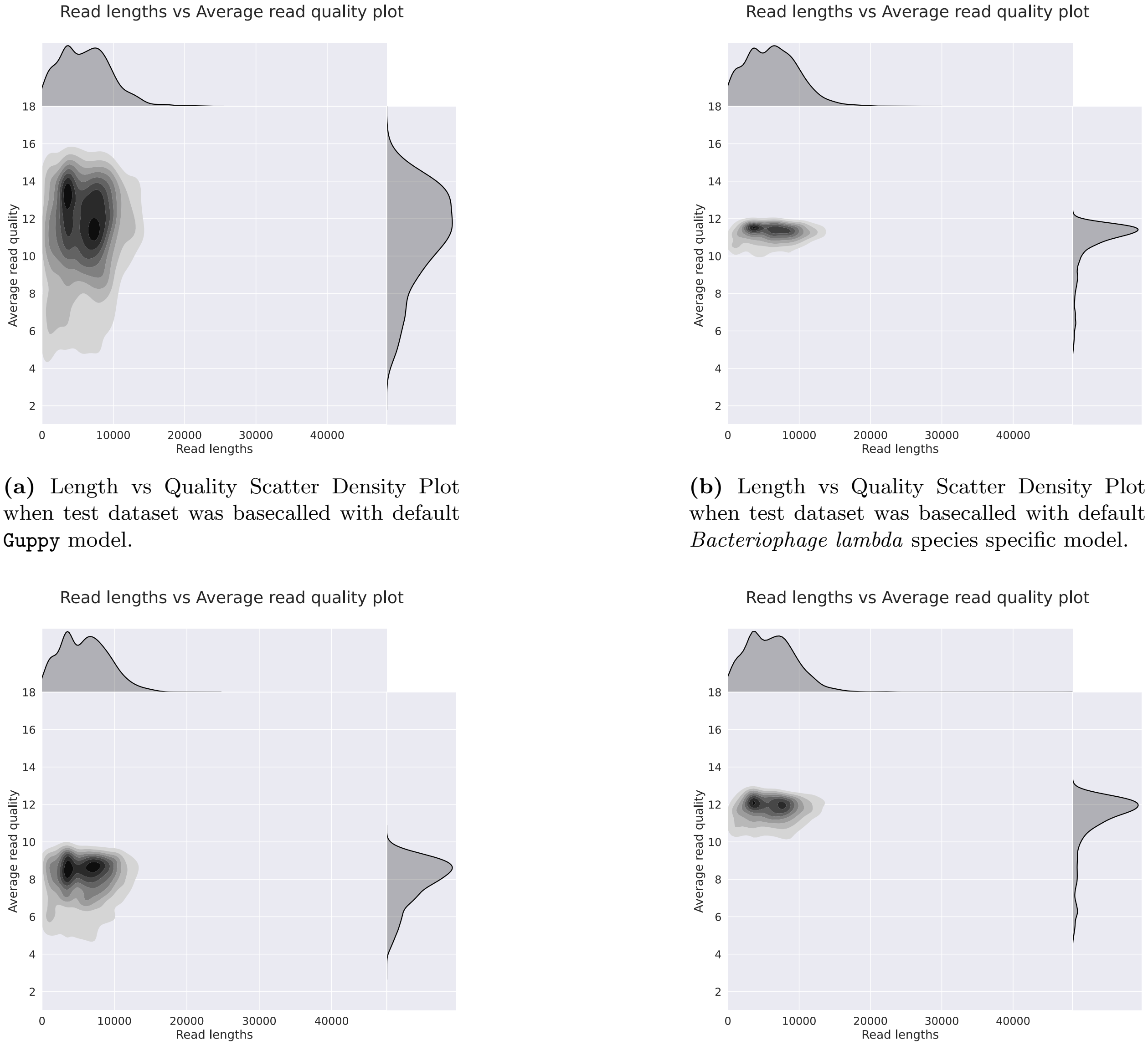
**(a)** Percent Identity vs Aligned Read Length when test dataset was basecalled with default `Guppy` model.

**(b)** Percent Identity vs Aligned Read Length when test dataset was basecalled with default *Bacteriophage lambda* species specific model.

**(c)** Percent Identity vs Aligned Read Length when test dataset was basecalled with `Taiyaki Walk-through` data model.

**(d)** Percent Identity vs Aligned Read Length when test dataset was basecalled with denoised model.

**Figure 5.6:** Comparison of Percent Identity vs Aligned Read Length between default and denoised models. All plots were generated by `NanoPlot`.



**(a)** Length vs Quality Scatter Density Plot when test dataset was basecalled with default `Guppy` model.

**(b)** Length vs Quality Scatter Density Plot when test dataset was basecalled with default *Bacteriophage lambda* species specific model.

**(c)** Length vs Quality Scatter Density Plot when test dataset was basecalled with `Taiyaki Walk-through` data model.

**(d)** Length vs Quality Scatter Density Plot when test dataset was basecalled with denoised model.

**Figure 5.7:** Comparison in Length vs Read Quality between default and denoised models. All plots were generated by `NanoPlot`.

# 6 Discussion

## 6.1 Discussion

For this thesis, custom models were created for nanopore basecalling using raw training dataset as well as denoised training datasets. These models were tested with raw testing dataset, as well as denoised testing datasets. The results indicate that certain denoised models performed best, and did so when tested against the raw signal. This implies that the use of wavelets can be introduced in the training and model generation processes, but there is no need to introduce wavelet analysis in any of the later stages such as: while generating data from a device during an experiment, sequencing that data, or any subsequent stages. This finding will save a lot of time and resources for academic and commercial experiments, where researchers can rely on using a custom model for more accurate basecalls and not be concerned with spending their limited computational resources in using wavelet analysis for any other parts of the experiment.

Custom denoised models generated by using a single organism outperformed others that were generated using multiple organisms in the training dataset, as reflected in Table B.1. This could be due to the difference in sizes, GC content and homopolyers in the genomes of species that were used to create multi-species models. The fold coverage of the involved datasets against their reference genomes were very different, and we believe it contributed to the difference in accuracy of custom models across different datasets. More research is needed to determine the underlying cause of this finding. Keeping the denoising parameters constant and using subsets of data with various fold coverage could present an explanation of the impact that fold coverage has on generating custom models.

Overall, the results from the experiments in this thesis work show that the reduction of noise in nanopore signal contributes to an increase in basecalling accuracy for the `Guppy` basecaller. An increase in consistency in the mean percentage identity and the mean read quality of our basecalls are an indicator that removing the noise from raw nanopore signal reduces the number of outliers and makes the basecalls consistent in accuracy and quality. An increase in the quality of single molecule basecalls impacts nanopore sequencing's applications and the bioinformatics community.

This thesis contributes to the bioinformatics and nanopore sequencing space in the following ways:

1. Efficacy of pre-filtering nanopore data: This thesis demonstrated the positive effect of pre-filtering and pre-processing of nanopore data using wavelet analysis techniques, and the resulting improvement in read accuracy and read quality from models trained on filtered data.

2. Choice of wavelet and parameters: After examining many potential wavelets empirically, the haar wavelet was found to consistently outperform other wavelet families. We believe it turned out to be the most suitable wavelet for this thesis due to its shape that closely resembles the expected shape of nanopore signal.

3. Denoising workflow: A workflow is provided for the bioinformatics community to follow for similar experiments for other species. The workflow can also be used for experiments unrelated to nanopore sequencing. Simple approaches were used for shortlisting the wavelet candidates and denoising parameters, as well as existing libraries built in Python like PyWavelets and SciPy.

4. Custom models: The models created in this thesis are provided for the bioinformatics community to use, and the workflow can be used as a guide to generate other models. An example of the workflow and models created for this project are provided at: https://github.com/coadunate/AWAND

The results obtained during the experiments for this thesis confirmed our initial hypothesis that removing noise in raw nanopore signals leads to better basecalling models. The increase in mean percentage identity and mean read quality obtained by using custom denoised models is a step towards new research on ways to reduce noise in raw nanopore signal using software and signal processing techniques. With more research in this area, nanopore sequencing can overcome the lower accuracy limitation it faces when compared to second generation sequencing methods. Increase in basecalling accuracy and quality can lead to nanopore sequencing being the future of sequencing for rapid real-time sequencing applications.

The scale of our findings is relevant to the nanopore sequencing community specifically, and sequencing community in general. Our approach can be used to improve basecalling for current nanopore sequencing users and can also encourage others to try nanopore sequencing, bringing it a wider range of academic and commercial users. Our approach can also be used as a guide for others interested in denoising nanopore sequencing data for basecalling and analysis. Our approach can also be used by the developers of basecallers that provide generic models with their basecalling toolkits. An overview of the wavelet analysis technique that worked to reduce noise in nanopore signal is provided in this thesis, which can be enhanced with exploration of a wider parameter space and different choices of wavelets.

To our knowledge, all software components used in this thesis can be updated to their latest versions in future without affecting the experiment pipeline. We also assume that if a major version upgrade is expected or support for software halts for any tools involved in the Taiyaki model generation process, ONT will provide a modified workflow for Taiyaki.

## 6.2 Future Work

This thesis used the *Bacteriophage lambda* genome which is a relatively simple and small genome to shortlist the wavelet space and parameters for denoising raw nanopore signal. The wavelet denoising process was tested first on *Bacteriophage lambda*, and then Cattle, Yeast, *E. coli* and Human datasets. The work in this thesis can be extended further to test our theory for a wider range of organisms with more complex genomes as well as other small organisms. The time required to train a model depends on the system specifications, GPU/CPU settings for `Guppy` and `Taiyaki`, and the size of reference genome(s). We suggest using our method for picking the optimal parameters for wavelet denoising any given dataset. Optimal parameters may vary across different sizes and complexities of genomes due to the nature of nanopore signal for different species, frequency of homopolymers, GC content, and contributors of noise and experimental setup. The custom multi-species models did not perform as well as single-species models, so additional work needs to be done in creating multi-species models and to determine the appropriate denoising parameters. The wavelet and parameter space was shortlisted using estimate increase in SNR and only the combinations of those shortlisted parameters were used to create customized models. Future work can test the entire parameter space or utilize different techniques for shortlisting. It will be interesting to see if denoising the nanopore signal and resulting change in basecall accuracy has a correlation with the genome size, G+C content, homopolymers, repetitive regions or organisms belonging to plant and animal kingdoms. Other basecallers can be tested with our models to see how that affects their performance and accuracy, compared to their default configuration. Forward and reverse strands were not separated or separately evaluated for this project and we relied on `NanoPack` and `minimap2` to deal with these conditions during analysis. Future work can look into the differences and similarities between the basecalled sequences in forward and reverse strands separately. Future work could also include looking at the effect of wavelet denoising on nanopore sequencing for ribonucleic acid (RNA). We believe our method of wavelet denoising will help create more accurate models and contribute to better basecalling for bioinformaticians and biologists.

# 7 Conclusion

The work in this thesis connected denoising techniques of signal processing to an application in DNA sequencing in the bioinformatics space. We proved our hypothesis that removing noise from a nanopore signal and creating a custom model with denoised signal before basecalling can increase the accuracy of basecalls. It was observed that custom models created using denoised nanopore signal enabled the basecaller to produce better basecalls due to a lesser chance of confusing noise disturbances with signal events. A limited parameter space was used in the experiments, which was shortlisted via analysis and observations made by us and other researchers. An increase of 5.3% was achieved in mean percentage identity while maintaining the mean read quality of basecalls for *Bacteriophage lambda* nanopore data.

The effect of wavelet pre-filtering on basecalls for nanopore sequencing was demonstrated in this thesis. An increase in mean percentage identity was observed for species specific models while keeping the mean read quality constant. However, pre-filtering applied on *Bos taurus* dataset had a negative effect on the accuracy and quality of basecalls. We assume that this negative effect on basecalling was due to the low coverage of the reference genome by the available dataset. In both cases, wavelet analysis seems to have a significant impact on the accuracy of basecalling for nanopore sequencing data. We believe that with more trials for different species we can determine the optimal denoising parameters for various species and multi-species models.

# References

[1] Maclyn McCarty. Discovering genes are made of DNA. *Nature*, 421(6921):406–406, January 2003.

[2] Anne Sayre. *Rosalind Franklin and DNA*. W. W. Norton & Company, New York, 1st edition edition, July 2000.

[3] J. D. Watson and F. H. C. Crick. Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid. *Nature*, 171(4356):737–738, April 1953. Number: 4356 Publisher: Nature Publishing Group.

[4] A. M. Maxam and W. Gilbert. A new method for sequencing DNA. *Proceedings of the National Academy of Sciences*, 74(2):560–564, February 1977. Publisher: National Academy of Sciences Section: Research Article.

[5] F. Sanger, S. Nicklen, and A. R. Coulson. DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences*, 74(12):5463–5467, December 1977. Publisher: National Academy of Sciences Section: Biological Sciences: Biochemistry.

[6] Michael L. Metzker. Sequencing in real time. *Nature Biotechnology*, 27(2):150–151, February 2009. Number: 2 Publisher: Nature Publishing Group.

[7] James M. Heather and Benjamin Chain. The sequence of sequencers: The history of sequencing DNA. *Genomics*, 107(1):1–8, January 2016.

[8] John Eid, Adrian Fehr, Jeremy Gray, Khai Luong, John Lyle, Geoff Otto, Paul Peluso, David Rank, Primo Baybayan, Brad Bettman, Arkadiusz Bibillo, Keith Bjornson, Bidhan Chaudhuri, Frederick Christians, Ronald Cicero, Sonya Clark, Ravindra Dalal, Alex deWinter, John Dixon, Mathieu Foquet, Alfred Gaertner, Paul Hardenbol, Cheryl Heiner, Kevin Hester, David Holden, Gregory Kearns, Xiangxu Kong, Ronald Kuse, Yves Lacroix, Steven Lin, Paul Lundquist, Congcong Ma, Patrick Marks, Mark Maxham, Devon Murphy, Insil Park, Thang Pham, Michael Phillips, Joy Roy, Robert Sebra, Gene Shen, Jon Sorenson, Austin Tomaney, Kevin Travers, Mark Trulson, John Vieceli, Jeffrey Wegener, Dawn Wu, Alicia Yang, Denis Zaccarin, Peter Zhao, Frank Zhong, Jonas Korlach, and Stephen Turner. Real-Time DNA Sequencing from Single Polymerase Molecules. *Science*, 323(5910):133–138, January 2009. Publisher: American Association for the Advancement of Science Section: Report.

[9] S. Howorka, S. Cheley, and H. Bayley. Sequence-specific detection of individual DNA strands using engineered nanopores. *Nature Biotechnology*, 19(7):636–639, July 2001.

[10] Anna L. McNaughton, Hannah E. Roberts, David Bonsall, Mariateresa de Cesare, Jolynne Mokaya, Sheila F. Lumley, Tanya Golubchik, Paolo Piazza, Jacqueline B. Martin, Catherine de Lara, Anthony Brown, M. Azim Ansari, Rory Bowden, Eleanor Barnes, and Philippa C. Matthews. Illumina and Nanopore methods for whole genome sequencing of hepatitis B virus (HBV). *Scientific Reports*, 9(1):7081, May 2019. Number: 1 Publisher: Nature Publishing Group.

[11] Daniel Branton and David W Deamer. *Nanopore Sequencing: An Introduction*. World Scientific Publishing Company, Singapore, SINGAPORE, 2019.

[12] Clive G. Brown and James Clarke. Nanopore development at Oxford Nanopore. *Nature Biotechnology*, 34(8):810–811, August 2016. Number: 8 Publisher: Nature Publishing Group.

[13] Daniel Branton, David W Deamer, Andre Marziali, Hagan Bayley, Steven A Benner, Thomas Butler, Massimiliano Di Ventra, Slaven Garaj, Andrew Hibbs, Xiaohua Huang, Stevan B Jovanovich, Predrag S Krstic, Stuart Lindsay, Xinsheng Sean Ling, Carlos H Mastrangelo, Amit Meller, John S Oliver, Yuriy V Pershin, J Michael Ramsey, Robert Riehn, Gautam V Soni, Vincent Tabard-Cossa, Meni Wanunu, Matthew Wiggin, and Jeffery A Schloss. The potential and challenges of nanopore sequencing. *Nature biotechnology*, 26(10):1146–1153, October 2008.

[14] Richard M. Leggett and Matthew D. Clark. A world of opportunities with nanopore sequencing. *Journal of Experimental Botany*, 68(20):5419–5429, November 2017. Publisher: Oxford Academic.

[15] Alberto Magi, Roberto Semeraro, Alessandra Mingrino, Betti Giusti, and Romina D'Aurizio. Nanopore sequencing data analysis: state of the art, applications and challenges. *Briefings in Bioinformatics*, 19(6):1256–1272, November 2018. Publisher: Oxford Academic.

[16] Kary B. Mullis, Francois Ferre, and Richard A. Gibbs, editors. *The Polymerase Chain Reaction*. Birkhäuser Basel, 1994.

[17] Shanika L. Amarasinghe, Shian Su, Xueyi Dong, Luke Zappia, Matthew E. Ritchie, and Quentin Gouil. Opportunities and challenges in long-read sequencing data analysis. *Genome Biology*, 21(1):30, February 2020.

[18] Alexander S. Mikheyev and Mandy M. Y. Tin. A first look at the Oxford Nanopore MinION sequencer. *Molecular Ecology Resources*, 14(6):1097–1102, 2014. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/1755-0998.12324.

[19] Lauren M. Petersen, Isabella W. Martin, Wayne E. Moschetti, Colleen M. Kershaw, and Gregory J. Tsongalis. Third-Generation Sequencing in the Clinical Laboratory: Exploring the Advantages and Challenges of Nanopore Sequencing. *Journal of Clinical Microbiology*, 58(1), December 2019.

[20] Tiantian Xiao and Wenhao Zhou. The third generation sequencing: the advanced approach to genetic diseases. *Translational Pediatrics*, 9(2):163–173, April 2020.

[21] Edward J Fox, Kate S Reid-Bayliss, Mary J Emond, and Lawrence A Loeb. Accuracy of Next Generation Sequencing Platforms. *Next generation, sequencing & applications*, 1, 2014.

[22] Andrea D. Tyler, Laura Mataseje, Chantel J. Urfano, Lisa Schmidt, Kym S. Antonation, Michael R. Mulvey, and Cindi R. Corbett. Evaluation of Oxford Nanopore's MinION Sequencing Device for Microbial Whole Genome Sequencing Applications. *Scientific Reports*, 8(1):10931, July 2018. Number: 1 Publisher: Nature Publishing Group.

[23] Christopher R. O'Donnell. Error Analysis and Parameter Estimation for Nanopore Based Molecular Detection. Master's thesis, University of California, Santa Cruz, United States – California, 2018. ISBN: 9780355865080.

[24] Eric Marinier, Daniel G. Brown, and Brendan J. McConkey. Pollux: platform independent error correction of single and mixed genomes. *BMC Bioinformatics*, 16(1):10, January 2015.

[25] Mircea Cretu Stancu, Markus J. van Roosmalen, Ivo Renkens, Marleen M. Nieboer, Sjors Middelkamp, Joep de Ligt, Giulia Pregno, Daniela Giachino, Giorgia Mandrile, Jose Espejo Valle-Inclan, Jerome Korzelius, Ewart de Bruijn, Edwin Cuppen, Michael E. Talkowski, Tobias Marschall, Jeroen de Ridder, and Wigard P. Kloosterman. Mapping and phasing of structural variation in patient genomes using nanopore sequencing. *Nature Communications*, 8(1):1326, November 2017. Number: 1 Publisher: Nature Publishing Group.

[26] Franka J. Rang, Wigard P. Kloosterman, and Jeroen de Ridder. From squiggle to basepair: computational approaches for improving nanopore sequencing read accuracy. *Genome Biology*, 19(1):90, July 2018.

[27] Matthew Puster. Improving the Signal-to-Noise of Nanopore Sensors. *Publicly Accessible Penn Dissertations*, January 2015.

[28] Raj D. Maitra, Jungsuk Kim, and William B. Dunbar. Recent advances in nanopore sequencing. *Electrophoresis*, 33(23):3418–3428, December 2012.

[29] Winston Timp, Jeffrey Comer, and Aleksei Aksimentiev. DNA Base-Calling from a Nanopore Using a Viterbi Algorithm. *Biophysical Journal*, 102(10):L37–L39, May 2012.

[30] Ryan R. Wick, Louise M. Judd, and Kathryn E. Holt. Performance of neural network basecalling tools for Oxford Nanopore sequencing. *Genome Biology*, 20(1):129, June 2019.

[31] Nicholas J. Loman, Joshua Quick, and Jared T. Simpson. A complete bacterial genome assembled de novo using only nanopore sequencing data. *Nature Methods*, 12(8):733–735, August 2015. Number: 8 Publisher: Nature Publishing Group.

[32] Sergey Koren, Brian P. Walenz, Konstantin Berlin, Jason R. Miller, Nicholas H. Bergman, and Adam M. Phillippy. Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation. *Genome Research*, 27(5):722–736, January 2017. Company: Cold Spring Harbor Laboratory Press Distributor: Cold Spring Harbor Laboratory Press Institution: Cold Spring Harbor Laboratory Press Label: Cold Spring Harbor Laboratory Press Publisher: Cold Spring Harbor Lab.

[33] R. A. Cottis, A. M. Homborg, and J. M. C. Mol. The relationship between spectral and wavelet techniques for noise analysis. *Electrochimica Acta*, 202:277–287, June 2016.

[34] S. Ayhan, S. Scherr, A. Bhutani, B. Fischbach, M. Pauli, and T. Zwick. Impact of Frequency Ramp Nonlinearity, Phase Noise, and SNR on FMCW Radar Accuracy. *IEEE Transactions on Microwave Theory and Techniques*, 64(10):3290–3301, October 2016. Conference Name: IEEE Transactions on Microwave Theory and Techniques.

[35] Wouter De Coster, Svenn D'Hert, Darrin T. Schultz, Marc Cruts, and Christine Van Broeckhoven. NanoPack: visualizing and processing long-read sequencing data. *Bioinformatics*, 34(15):2666–2669, August 2018. Publisher: Oxford Academic.

[36] Matei David, L. J. Dursi, Delia Yao, Paul C. Boutros, and Jared T. Simpson. Nanocall: an open source basecaller for Oxford Nanopore sequencing data. *Bioinformatics*, 33(1):49–55, January 2017. Publisher: Oxford Academic.

[37] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, ICML '06, pages 369–376, New York, NY, USA, June 2006. Association for Computing Machinery.

[38] Nick Vereecke, Jade Bokma, Freddy Haesebrouck, Hans Nauwynck, Filip Boyen, Bart Pardon, and Sebastiaan Theuns. High quality genome assemblies of Mycoplasma bovis using a taxon-specific Bonito basecaller for MinION and Flongle long-read nanopore sequencing. *BMC Bioinformatics*, 21(1):517, November 2020.

[39] Haotian Teng, Minh Duc Cao, Michael B Hall, Tania Duarte, Sheng Wang, and Lachlan J M Coin. Chiron: translating nanopore raw signal directly into nucleotide sequence using deep learning. *GigaScience*, 7(giy037), May 2018.

[40] Vladimír Boža, Broňa Brejová, and Tomáš Vinař. DeepNano: Deep recurrent neural networks for base calling in MinION nanopore reads. *PLoS ONE*, 12(6), June 2017.

[41] Heng Li. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, 34(18):3094–3100, September 2018. Publisher: Oxford Academic.

[42] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, October 1990.

[43] Alessio Fragasso, Sonja Schmid, and Cees Dekker. Comparing Current Noise in Biological and Solid-State Nanopores. *ACS Nano*, 14(2):1338–1349, February 2020.

[44] R. M. M. Smeets, U. F. Keyser, N. H. Dekker, and C. Dekker. Noise in solid-state nanopores. *Proceedings of the National Academy of Sciences*, 105(2):417–421, January 2008. Publisher: National Academy of Sciences Section: Physical Sciences.

[45] Shengfa Liang, Feibin Xiang, Zifan Tang, Reza Nouri, Xiaodong He, Ming Dong, and Weihua Guan. Noise in nanopore sensors: Sources, models, reduction, and benchmarking. *Nanotechnology and Precision Engineering*, 3(1):9–17, March 2020.

[46] Will Gragido, Johnl Pirc, Nick Selby, and Daniel Molina. Chapter 4 - Signal-to-Noise Ratio. In Will Gragido, Johnl Pirc, Nick Selby, and Daniel Molina, editors, *Blackhatonomics*, pages 45–55. Syngress, Boston, January 2013.

[47] Chen-Chi Chien. *Improving Signal to Noise Ratio and Time Resolution for Solid-state Nanopore Measurements*. Ph.D., University of Pennsylvania, United States – Pennsylvania, 2018. ISBN: 9780438403598.

[48] Adrian Constantin. *Fourier Analysis: Volume 1: Theory*, volume 1 of *London Mathematical Society Student Texts*. Cambridge University Press, Cambridge, 2016.

[49] K. R. Rao, D. N. Kim, and J. J. Hwang. Applications. In K.R. Rao, D.N. Kim, and J.-J. Hwang, editors, *Fast Fourier Transform - Algorithms and Applications*, Signals and Communication Technology, pages 235–316. Springer Netherlands, Dordrecht, 2010.

[50] Mawardi Bahri, Eckhard S. M. Hitzer, Ryuichi Ashino, and Rémi Vaillancourt. Windowed Fourier transform of two-dimensional quaternionic signals. *Applied Mathematics and Computation*, 216(8):2366–2379, June 2010.

[51] Christopher Heil. Ten Lectures on Wavelets (Ingrid Daubechies). *SIAM Review*, 35(4):666–669, December 1993. Publisher: Society for Industrial and Applied Mathematics.

[52] A. A. Cardoso and F. H. T. Vieira. Adaptive estimation of Haar wavelet transform parameters applied to fuzzy prediction of network traffic. *Signal Processing*, 151:155–159, October 2018.

[53] Dmitry Popov, Artem Gapochkin, and Alexey Nekrasov. An Algorithm of Daubechies Wavelet Transform in the Final Field When Processing Speech Signals. *Electronics*, 7(7):120, July 2018. Number: 7 Publisher: Multidisciplinary Digital Publishing Institute.

[54] A. Zaeni, T. Kasnalestari, and U. Khayam. Application of Wavelet Transformation Symlet Type and Coiflet Type For Partial Discharge Signals Denoising. In *2018 5th International Conference on Electric Vehicular Technology (ICEVT)*, pages 78–82, October 2018.

[55] Hong Liu, Lin-pei Zhai, Ying Gao, Wen-ming Li, and Jiu-fei Zhou. Image compression based on biorthogonal wavelet transform. In *IEEE International Symposium on Communications and Information Technology, 2005. ISCIT 2005.*, volume 1, pages 598–601, October 2005.

[56] Kholkhal Mourad and Bereksi Reguig Fethi. Efficient automatic detection of QRS complexes in ECG signal based on reverse biorthogonal wavelet decomposition and nonlinear filtering. *Measurement*, 94:663–670, December 2016.

[57] Xiaoyan Wen, Dongsheng Zhang, Yu Qian, Jieyan Li, and Nie Fei. Improving the peak wavelength detection accuracy of Sn-doped, H2-loaded FBG high temperature sensors by wavelet filter and Gaussian curve fitting. *Sensors and Actuators A: Physical*, 174:91–95, February 2012.

[58] I. Daubechies. The wavelet transform, time-frequency localization and signal analysis. *IEEE Transactions on Information Theory*, 36(5):961–1005, September 1990. Conference Name: IEEE Transactions on Information Theory.

[59] Christopher Torrence and Gilbert P. Compo. A Practical Guide to Wavelet Analysis. *Bulletin of the American Meteorological Society*, 79(1):61–78, January 1998. Publisher: American Meteorological Society.

[60] D.L. Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41(3):613–627, May 1995. Conference Name: IEEE Transactions on Information Theory.

[61] Mehdi Kchouk, Jean-Francois Gibrat, and Mourad Elloumi. Generations of Sequencing Technologies: From First to Next Generation. *Biology and Medicine*, 09, January 2017.

[62] Roger Volden, Theron Palmer, Ashley Byrne, Charles Cole, Robert J. Schmitz, Richard E. Green, and Christopher Vollmers. Improving nanopore read accuracy with the R2C2 method enables the sequencing of highly multiplexed full-length single-cell cDNA. *Proceedings of the National Academy of Sciences of the United States of America*, 115(39):9726–9731, September 2018.

[63] Hengyun Lu, Francesca Giordano, and Zemin Ning. Oxford Nanopore MinION Sequencing and Genome Assembly. *Genomics, Proteomics & Bioinformatics*, 14(5):265–279, October 2016.

[64] Rory Bowden, Robert W. Davies, Andreas Heger, Alistair T. Pagnamenta, Mariateresa de Cesare, Laura E. Oikkonen, Duncan Parkes, Colin Freeman, Fatima Dhalla, Smita Y. Patel, Niko Popitsch, Camilla L. C. Ip, Hannah E. Roberts, Silvia Salatino, Helen Lockstone, Gerton Lunter, Jenny C. Taylor, David Buck, Michael A. Simpson, and Peter Donnelly. Sequencing of human genomes with nanopore technology. *Nature Communications*, 10, April 2019.

[65] T. Laver, J. Harrison, P.A. O'Neill, K. Moore, A. Farbos, K. Paszkiewicz, and D.J. Studholme. Assessing the performance of the Oxford Nanopore Technologies MinION. *Biomolecular Detection and Quantification*, 3:1–8, March 2015.

[66] Anbo Zhou, Timothy Lin, and Jinchuan Xing. Evaluating nanopore sequencing data processing pipelines for structural variation identification. *Genome Biology*, 20(1):237, November 2019.

[67] Damla Senol Cali, Jeremie S. Kim, Saugata Ghose, Can Alkan, and Onur Mutlu. Nanopore sequencing technology and tools for genome assembly: computational analysis of the current state, bottlenecks and future directions. *Briefings in Bioinformatics*, 20(4):1542–1559, July 2019. Publisher: Oxford Academic.

[68] Kathryn Dumschott, Maximilian H.-W. Schmidt, Harmeet Singh Chawla, Rod Snowdon, and Björn Usadel. Oxford Nanopore sequencing: new opportunities for plant genomics? *Journal of Experimental Botany*, 71(18):5313–5322, September 2020. Publisher: Oxford Academic.

[69] Juliane C Dohm, Philipp Peters, Nancy Stralis-Pavese, and Heinz Himmelbauer. Benchmarking of long-read correction methods. *NAR Genomics and Bioinformatics*, 2(lqaa037), June 2020.

[70] Chao-Chen Chen and Fuchiang Rich Tsui. Comparing different wavelet transforms on removing electrocardiogram baseline wanders and special trends. *BMC Medical Informatics and Decision Making*, 20(11):343, December 2020.

[71] L. Xu and C. Huang. Wavelet-Based SNR Analysis in Building Satellite Terminal Fault Identification System. In *2008 IEEE International Conference on Communications*, pages 1942–1946, May 2008. ISSN: 1938-1883.

[72] G. M. Aldonin, A. V. Soldatov, and V. V. Cherepanov. Wavelet Analysis of Cardiac Electrical Activity Signals. *Biomedical Engineering*, 52(2):120–124, July 2018.

[73] Siddharth Shekar, Chen-Chi Chien, Andreas Hartel, Peijie Ong, Oliver B. Clarke, Andrew Marks, Marija Drndic, and Kenneth L. Shepard. Wavelet Denoising of High-Bandwidth Nanopore and Ion-Channel Signals. *Nano Letters*, 19(2):1090–1097, 2019.

[74] Gregory R. Lee, Ralf Gommers, Filip Waselewski, Kai Wohlfahrt, and Aaron O'Leary. PyWavelets: A Python package for wavelet analysis. *Journal of Open Source Software*, 4(36):1237, April 2019.

[75] Damian Gogolewski, Włodzimierz Makieła, Krzysztof Stepień, Paweł Zmarzły, and Mateusz Wrzochal. The Assessment of Wavelet Transform Parameters Regarding Its Use in 3d Surface Filtering. *Annals of DAAAM & Proceedings*, 29:1191–1196, January 2018. Publisher: DAAAM International.

[76] A. I. Nazmutdinova and V. N. Milich. Dependence of the results of classification of multispectral images of forest vegetation on wavelet-transform parameters. *Optoelectronics, Instrumentation and Data Processing*, 52(3):231–237, May 2016.

[77] Sherwood R. Casjens and Roger W. Hendrix. Bacteriophage lambda: early pioneer and still relevant. *Virology*, 0:310–330, May 2015.

[78] C. Hillis, HA Lardner, and MG Links. Using Nanopore Technology to Determine Paternity in Multi-sire Breeding Pastures., 2019.

[79] P. Welch. The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics*, 15(2):70–73, June 1967. Conference Name: IEEE Transactions on Audio and Electroacoustics.

[80] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C. J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, and Paul van Mulbregt. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272, March 2020. Bandiera_abtest: a Cc_license_type: cc_by Cg_type: Nature Research Journals Number: 3 Primary_atype: Reviews Publisher: Nature Publishing Group Subject_term: Biophysical chemistry;Computational biology and bioinformatics;Technology Subject_term_id: biophysical-chemistry;computational-biology-and-bioinformatics;technology.

[81] Pan Qi, Slavisa Jovanovic, Jinmi Lezama, and Patrick Schweitzer. Discrete wavelet transform optimal parameters estimation for arc fault detection in low-voltage residential power networks. *Electric Power Systems Research*, 143:130–139, February 2017.

[82] M. Bigerelle, G. Guillemot, Z. Khawaja, M. El Mansori, and J. Antoni. Relevance of Wavelet Shape Selection in a complex signal. *Mechanical Systems and Signal Processing*, 41(1):14–33, December 2013.

# Appendix A

# Wavelet Denoising Example

```
import sys
from os import listdir
from os.path import isfile, join
from shutil import copyfile
from datetime import datetime
from collections import defaultdict, namedtuple
import hashlib
import pywt
import h5py
import numpy as np

def denoise(data, wavelet, threshold, maxlev):
    ''' remove noise from signal using specified parameters '''

    w = pywt.Wavelet(wavelet)

    # Decompose into wavelet components, to the level selected:
    coeffs = pywt.wavedec(data, wavelet, level=maxlev)

    for i in range(1, len(coeffs)):
        coeffs[i] = pywt.threshold(coeffs[i], threshold*max(coeffs[i]), mode='
            soft')

    datarec = pywt.waverec(coeffs, wavelet)

    return datarec

def file_as_bytes(file):
    ''' to be used for checksum '''
    with file:
        return file.read()


if __name__ == ”__main__”:
    ''' extract signal from all files in a directory and remove noise '''

    # more options for wavelets, thresholds and levels can be added to the
        respective arrays
    wavelets = ['haar']
    thresholds = [0.04]
    levels = [4]

    fast5_directory = sys.argv[1]
    output_directory = fast5_directory+”/denoised_files–”+str(datetime.now(tz=
        None))+”/”
```

```
files = [f for f in listdir(fast5_directory) if isfile(join(
    fast5_directory, f))]

os.mkdir(output_directory)

for src in files:

    dst = output_directory+os.path.splitext(src)[0]

    # create a checksum to verify the validity of denoised file
    hash = str(hashlib.md5(file_as_bytes(open(fast5_directory+”/”+src, 'rb
        '))).hexdigest())

    hdf = h5py.File(fast5_directory+”/”+src, 'r+')

    fast5_info = hdf['UniqueGlobalKey/channel_id'].attrs
    channelInfo = namedtuple(
        'channelInfo',
        ('offset', 'range', 'digitisation', 'number', 'sampling_rate')
    )

    channel_info = channelInfo(
        fast5_info['offset'], fast5_info['range'],
        fast5_info['digitisation'], fast5_info['channel_number'],
        fast5_info['sampling_rate'].astype('int_'))

    shift, scale = (-1 * channel_info.offset, channel_info.digitisation /
        channel_info.range)

    keys = list(hdf['Raw/Reads'].keys())
    signal = hdf['Raw/Reads/'][keys[0]]['Signal']
    signal_duration = hdf['Raw/Reads/'][keys[0]].attrs['duration']

    signal = np.array(signal)

    signal_n = (signal - shift) / scale

    hdf.close()

    # apply wavelet denoising to signal with all parameter combinations
    for wavelet in wavelets:
        for threshold in thresholds:
            for level in levels:

                path = dst+”-”+wavelet+”-”+str(threshold)+”-”+str(level)+”
                    .fast5”
                copyfile(fast5_directory+”/”+src, path)

                hdf = h5py.File(path, 'r+')
                keys = list(hdf['Raw/Reads'].keys())

                dataset = hdf['Raw/Reads/'+keys[0]]
```

```
del hdf[”Raw/Reads/”+keys[0]+”/Signal”]

denoised_n = denoise(signal_n, wavelet, threshold, level)
denoised = np.array((denoised_n*scale) + shift, dtype=”
    int16”)[:signal_duration]
np.reshape(denoised, (len(signal), 1))
print(denoised.shape)

# write the denoised signal to file
hdf.create_dataset(”Raw/Reads/”+keys[0]+”/Signal”, data=
    denoised, maxshape=(None,))
hdf.close()
```

# Appendix B

# All Models and Experiments

Table B.1 shows a complete list of models that were created and used in this thesis. All models were created using `Taiyaki` with the exception of the first row which came from `Guppy`.

| Test dataset | Model | Coverage w.r.t Reference Genome | Mean Percentage Identity | Mean Quality | Median Percentage Identity | Median Quality |
|---|---|---|---|---|---|---|
| Raw *Bacteriophage lambda* signal | Default `Guppy` model | | 90.60 | 11.00 | 92.60 | 11.50 |
| Raw *Bacteriophage lambda* signal | *Bacteriophage lambda* Model | 4117 | 96.20 | 10.60 | 98.40 | 11.10 |
| Raw *Bacteriophage lambda* signal | *Bacteriophage lambda* training dataset/2 | 2055 | 93.90 | 10.70 | 95.90 | 11.10 |
| Raw *Bacteriophage lambda* signal | *Bacteriophage lambda* training dataset/4 | 1031 | 96.00 | 10.00 | 98.30 | 10.50 |
| Raw *Bacteriophage lambda* signal | *Bacteriophage lambda* training dataset/8 | 509 | 95.30 | 9.60 | 97.60 | 10.00 |
| Raw *Bacteriophage lambda* signal | *Bacteriophage lambda* training dataset/16 | 256 | 93.80 | 10.00 | 96.00 | 10.30 |
| Raw *Bacteriophage lambda* signal | *Bacteriophage lambda* training dataset/32 | 124 | 90.40 | 10.20 | 92.10 | 10.40 |
| Raw *Bacteriophage lambda* signal | 1/2 denoised *Bacteriophage lambda* 4_004 + denoised `Taiyaki Walk-through` reads 4_004 | | 86.6 | 6.6 | 88.1 | 6.8 |
| Raw *Bacteriophage lambda* signal | 1/4th denoised *Bacteriophage lambda* 4_004 + denoised `Taiyaki Walk-through` reads 4_004 | | 86.50 | 6.90 | 88.00 | 7.10 |
| Raw *Bacteriophage lambda* signal | 1/8th denoised *Bacteriophage lambda* 4_004 + denoised `Taiyaki Walk-through` reads 4_004 | | 85.90 | 6.70 | 87.40 | 6.90 |
| Raw *Bacteriophage lambda* signal | denoised *Bacteriophage lambda* 5_003 | | 96.20 | 9.30 | 98.30 | 9.70 |
| Raw *Bacteriophage lambda* signal | denoised *Bacteriophage lambda* 3_004 | | 94.70 | 10.90 | 96.90 | 11.40 |
| Raw *Bacteriophage lambda* signal | denoised *Bacteriophage lambda* 4_004 | | 95.90 | 11.00 | 98.10 | 11.60 |
| Raw *Bacteriophage lambda* signal | denoised *Bacteriophage lambda* 5_004 | | 95.70 | 10.10 | 97.90 | 10.60 |
| 1_001 | denoised *Bacteriophage lambda* 1_001 | | 95.90 | 10.70 | 98.20 | 11.30 |
| 1_001 | *Bacteriophage lambda* Model | | 95.60 | 10.50 | 98.00 | 11.00 |
| 3_003 | denoised *Bacteriophage lambda* 3_003 | | 95.60 | 9.10 | 98.10 | 9.50 |

|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
| 4_003 | denoised *Bacteriophage lambda* 4_003 |  | 95.20 | 9.20 | 97.90 | 9.70 |
| 5_003 | denoised *Bacteriophage lambda* 5_003 |  | 95.20 | 9.00 | 97.90 | 9.50 |
| 5_003 | denoised *Bacteriophage lambda* 3_004 |  | 93.50 | 10.70 | 96.50 | 11.30 |
| 3_004 | denoised *Bacteriophage lambda* 3_004 |  | 93.40 | 10.80 | 96.70 | 11.40 |
| 3_004 | denoised *Bacteriophage lambda* 4_004 |  | 94.60 | 10.70 | 97.70 | 11.30 |
| 4_004 | denoised *Bacteriophage lambda* 4_004 |  | 94.40 | 10.60 | 97.60 | 11.30 |
| 5_004 | denoised *Bacteriophage lambda* 5_004 |  | 94.20 | 9.70 | 97.40 | 10.30 |
| 5_004 | denoised *Bacteriophage lambda* 1_001 |  | 91.00 | 9.50 | 95.00 | 10.10 |
| 3_006 | denoised *Bacteriophage lambda* 3_006 |  | 93.30 | 8.50 | 97.10 | 9.00 |
| 4_006 | denoised *Bacteriophage lambda* 4_006 |  | 92.60 | 8.90 | 96.80 | 9.50 |
| 5_006 | denoised *Bacteriophage lambda* 5_006 |  | 92.60 | 8.40 | 96.70 | 9.00 |
| 3_009 | denoised *Bacteriophage lambda* 3_009 |  | 91.30 | 8.90 | 95.60 | 9.50 |
| 4_009 | denoised *Bacteriophage lambda* 4_009 |  | 86.20 | 9.20 | 89.50 | 9.40 |
| 5_009 | denoised *Bacteriophage lambda* 5_009 |  | 86.40 | 9.50 | 89.00 | 9.80 |
| 1_001 | Default `Guppy` model |  | 88.70 | 10.30 | 91.20 | 10.60 |
| 3_003 | Default `Guppy` model |  | 83.30 | 8.60 | 85.90 | 8.80 |
| 4_003 | Default `Guppy` model |  | 82.70 | 8.40 | 85.10 | 8.60 |
| 5_003 | Default `Guppy` model |  | 82.60 | 8.40 | 85.00 | 8.50 |
| 3_004 | Default `Guppy` model |  | 81.20 | 8.00 | 83.10 | 8.10 |
| 4_004 | Default `Guppy` model |  | 80.30 | 7.70 | 82.00 | 7.80 |
| 5_004 | Default `Guppy` model |  | 80.10 | 7.70 | 81.80 | 7.80 |
| 3_006 | Default `Guppy` model |  | 76.70 | 6.90 | 76.90 | 6.90 |
| 4_006 | Default `Guppy` model |  | 75.20 | 6.70 | 75.20 | 6.60 |
| 5_006 | Default `Guppy` model |  | 75.00 | 6.60 | 75.20 | 6.60 |
| 3_009 | Default `Guppy` model |  | 71.60 | 6.10 | 70.90 | 6.00 |
| 4_009 | Default `Guppy` model |  | 70.40 | 5.90 | 69.90 | 5.80 |
| 5_009 | Default `Guppy` model |  | 70.2 | 5.8 | 69.8 | 5.8 |
| `Taiyaki Walk-through` | denoised *Bacteriophage lambda* 5_004 |  | 69.9 | 6.4 | 69.8 | 6.4 |
| Raw *Bacteriophage lambda* signal | `Taiyaki Walk-through` generic model |  | 89 | 7.8 | 90.8 | 8.1 |
| 5_004 | hybrid model created from Raw *Bacteriophage lambda* signal + denoised *Bacteriophage lambda* 5_004 |  | 90.7 | 9.3 | 95 | 9.9 |
| Raw Cattle signal | Denoised Cattle dataset 4_004 |  | 74.8 | 10.3 | 75.4 | 10.4 |

**Table B.1:** All experiment runs and their output for Percentage Identity and Read Quality

# Appendix C

# Supplementary Information and Figures

Figure C.3 shows the trend of percentage of matches across various test datasets and basecalling model combinations. The highest percentage matches can be seen at the peaks corresponding to raw signal and $4_0 04$ and $1_0 01$ models.

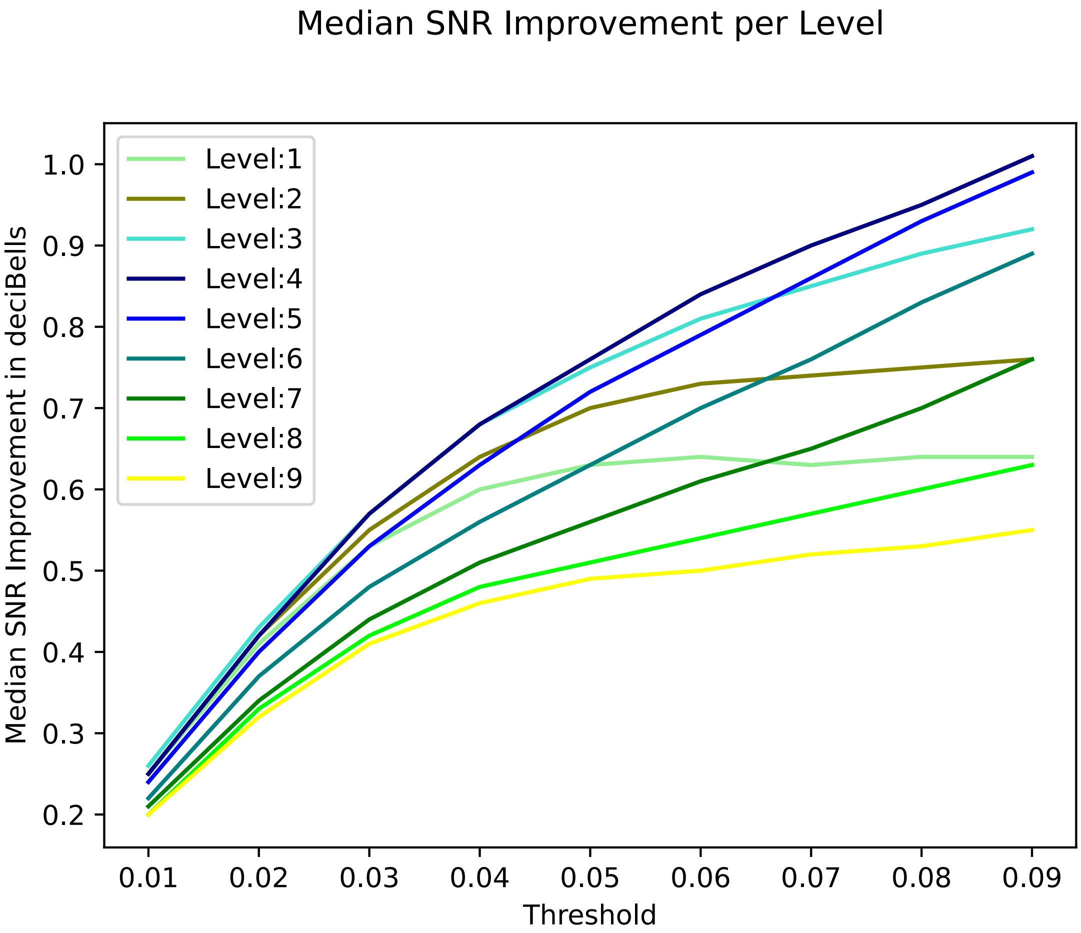


**Figure C.1:** Median SNR improvement for each level-threshold combination

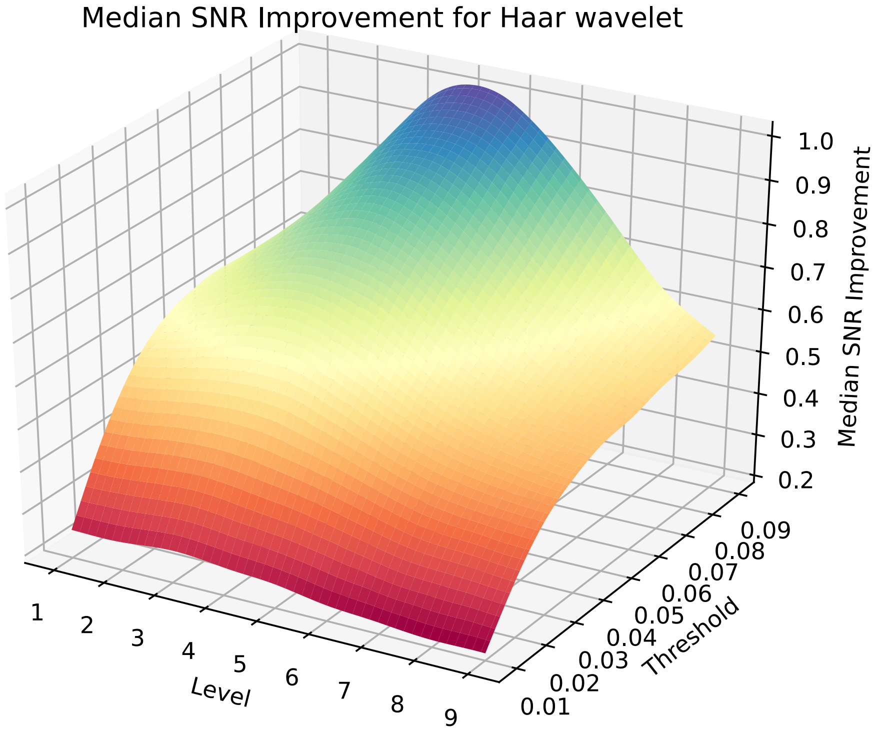


**Figure C.2:** Median SNR improvement for each level-threshold combination

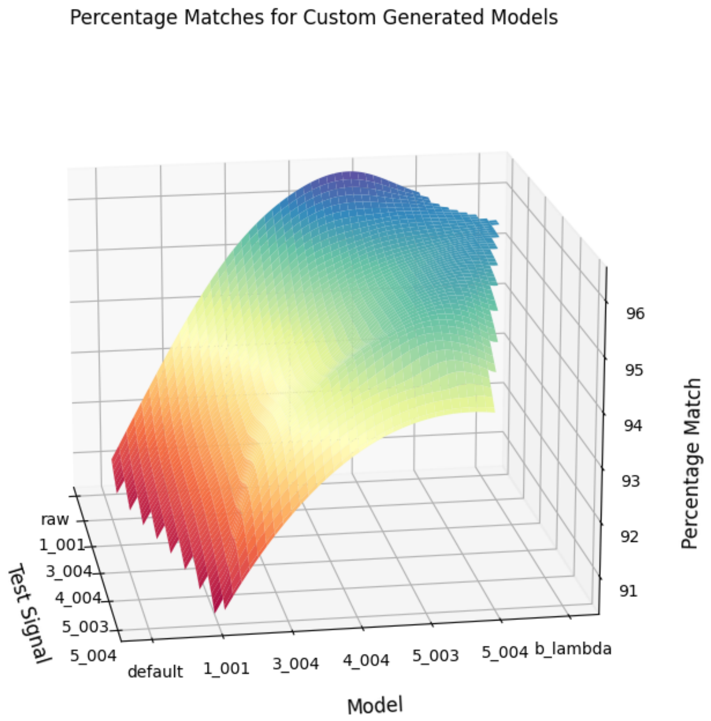


**Figure C.3:** Surface plot for percentage matches across testing datasets and model combinations

| **Decomposition Level** | **Threshold** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **0.01** | **0.02** | **0.03** | **0.04** | **0.05** | **0.06** | **0.07** | **0.08** | **0.09** |
| **1** | 0.25 | 0.41 | 0.53 | 0.6 | 0.63 | 0.64 | 0.63 | 0.64 | 0.64 |
| **2** | 0.25 | 0.42 | 0.55 | 0.64 | 0.7 | 0.73 | 0.74 | 0.75 | 0.76 |
| **3** | 0.26 | 0.43 | 0.57 | 0.68 | 0.75 | 0.81 | 0.85 | 0.89 | 0.92 |
| **4** | 0.25 | 0.42 | 0.57 | 0.68 | 0.76 | 0.84 | 0.9 | 0.95 | 1.01 |
| **5** | 0.24 | 0.4 | 0.53 | 0.63 | 0.72 | 0.79 | 0.86 | 0.93 | 0.99 |
| **6** | 0.22 | 0.37 | 0.48 | 0.56 | 0.63 | 0.7 | 0.76 | 0.83 | 0.89 |
| **7** | 0.21 | 0.34 | 0.44 | 0.51 | 0.56 | 0.61 | 0.65 | 0.7 | 0.76 |
| **8** | 0.2 | 0.33 | 0.42 | 0.48 | 0.51 | 0.54 | 0.57 | 0.6 | 0.63 |
| **9** | 0.2 | 0.32 | 0.41 | 0.46 | 0.49 | 0.5 | 0.52 | 0.53 | 0.55 |

**Table C.1:** Median SNR improvement for each level-threshold combination