2022

Bacterial Contamination in Public ATAC-Seq Data and Alignment-Free Detection Methods

Drake Michael Aesoph
West Virginia University, da00009@mix.wvu.edu

Recommended Citation
Aesoph, Drake Michael, "Bacterial Contamination in Public ATAC-Seq Data and Alignment-Free Detection Methods" (2022). *Graduate Theses, Dissertations, and Problem Reports*. 11566.
https://researchrepository.wvu.edu/etd/11566

Bacterial Contamination in Public ATAC-Seq Data and Alignment-Free Detection Methods

Drake Aesoph

A problem report submitted to the
Benjamin M. Statler College of Engineering and Mineral Resources
at West Virginia University

in partial fulfillment of the requirements for the degree of
Master of Science in
Computer Science

Donald Adjeroh, Ph.D., Chair
Gangqing Hu, Ph.D., Co-chair
Lan Guo, Ph.D.
Lane Department of Computer Science and Electrical Engineering
Morgantown, West Virginia
2022

Keywords: Mycoplasma; ATAC-seq; bacterial contamination; NGS; machine learning; bioinformatics tool

ABSTRACT

Bacterial Contamination in Public ATAC-Seq Data and Alignment-Free Detection Methods

Drake Aesoph

ATAC-seq is a new high-throughput sequencing technology for measuring chromatin accessibility within genomic samples. It can be used to discover new information about open regions, nucleosome positions, transcription factor binding sites, and DNA methylation. It is especially useful when combined with other next-generation sequencing techniques, such as RNA-seq. Unlike previous technologies, however, ATAC-seq is more sensitive to bacterial contamination, which is a well-known problem in cell cultures that can lead to incorrect experimental results. Previous studies have measured the contamination in public RNA-seq data and found that 5%-10% of samples were contaminated. In this report, we investigate the prevalence of contamination in ATAC-seq samples, rather than RNA-seq data, uploaded to the Sequence Read Archive using two popular alignment-based tools: Bowtie 2 and Kraken 2. We then develop an alignment-free method of detection using machine learning and a novel method of estimating DNA fragment lengths from paired-end ATAC-seq data. Our results show that around 5% of ATAC-seq samples are contaminated and our machine learning method is able to correctly classify 97% of samples as contaminated or not while using less computational resources than the alignment-based tools. Thus, our method shows promise as a preliminary rapid screening tool for contamination in labs with limited access huge to computational resources.

ACKNOWLEDGEMENTS

I would like to thank my advisors Dr. Gangqing Hu and Dr. Donald Adjeroh for their professional help and guidance while researching this topic. I would also like to thank Dr. Pyzdrowski for pointing the way to a path of research.

Contents

Chapter 1: Introduction

The first and arguably most important first step in working with data is verifying the quality of the data. Without this crucial step, any conclusions or discoveries made from analyses of the data may be completely invalid and pointless. Bioinformatics is a new and rapidly developing field, with hundreds of terabytes of data being produced from thousands of experiments through next generation sequencing (NGS). This data is often made publicly available for other researchers to freely reanalyze in new and interesting ways in their own research. One of the ways scientists publicly share their sequencing data is through the sequencing read archive (SRA) [1]. However, due to the incredibly high volume of data being uploaded to this service every day, quality control of the sequencing data relies on the lab that publishes the data.

Recent studies found that reads originating from bacterial sources in next generation sequencing data are widespread in the public data landscape [2], [3], with some surveys estimating as much as 5% of RNA-seq samples are contaminated [4]. It has previously been shown how unwanted contaminants can invalidate experiments by altering results in unexpected ways, as described in [5]. Given the fact that new bioinformatics technologies are constantly being developed or improved upon, the scientific community has to continually develop new methods of fighting contamination.

Previous studies on the prevalence of contamination in public data have only focused on RNA-seq, which made up the majority of data used in publications at the time. However, it is unclear if the pervasiveness of contamination in RNA-seq data also applies to newer technologies such as ATAC-seq, which may actually be more sensitive to contamination. We intend to perform a systematic scan for contamination on all available human ATAC-seq samples and compare results to existing scans on RNA-seq.

While scanning for contamination using current state of the art alignment-based tools, we found that they used a very large amount of computational resources. This was especially apparent given the extent of our study. Not all scientists have access to such high-performance computational resources. So, in order to help these scientists perform the crucial quality control step of checking for contamination on their own samples, we investigate an alignment-free method of contamination detection.

The aims of this project are to 1) survey the prevalence of contamination in the SRA database and 2) develop an alignment-free method for a fast detection of bacterial contamination.

Chapter 2: Background and Literature Review

2.1 The Genomic Landscape

DNA is an essential building block for all biology on earth. The DNA sequence consists of coding region for proteins which form the structure of our cells. To control the transcription of certain regions of genomic DNA into protein, the promoter region of a gene and enhancers need to be accessible to transcriptional machinery known as the RNA Polymerases, a process referred to as epigenetic regulation [6], [7]. To measure and quantify the accessibility of these regions, a variety of epigenetic assays based on deep sequencing are currently available to profile genome-wide landscape of chromatin accessibility including DNase-Seq, FAIRE-Seq, and ATAC-seq.

2.2 ATAC-seq

ATAC-seq (Assay for Transposase Chromatin) is a high-throughput sequencing technology developed to assay chromatin accessibility within genomes. Using ATAC-seq has facilitated the discovery of many features of the epigenetic landscape such as nucleosome positions and open regions. Compared to other methods, ATAC-seq is faster and more sensitive [8], [9]. ATAC-seq preferentially targets DNA sequences not protected by nucleosomes, for example active promoters, enhancers and mitochondrial DNAs. Also because of this feature, it is more sensitive to bacterial contaminants on the host cell or culture media [4].

2.3 Bacterial Contaminants in NGS

There are many steps and materials involved in cell culture, and each one is a potential source of contamination. In response to an infection by an external source, a cell may alter its internal state, leading to differential gene expression and confounding experimental results. The cell is also then competing for resources with any bacteria that may be attempting to grow alongside it. Since bacterial contamination is not an uncommon phenomenon in cell culture, existing literature has addressed its prevalence and introduce both molecular and computational tools for contamination detection [10], [11].

2.4 Prior Work on Computational Detection of Contamination

In [12], the authors discuss the presence of mycoplasma contamination via a survey of NCBI's RNA-seq archive which included 9395 samples from both rodents and primates. At the time, they found that 11% of the samples were contaminated. They then generated a generalized model and discovered 61 genes which were statistically associated with the mycoplasma-mapped read counts. The limitations of their paper, however, is that they only focused on RNA-seq libraries from cell cultures and only checked for mycoplasma, no other bacteria were considered. The majority of the RNA-Seq data targets messenger RNAs via polyA enrichment or ribosomal RNA depletion. Bacteria RNAs do not have polyA tails and most of the reads that map to mycoplasma are the very abundant ribosomal RNAs that escape the polyA enrichment.

In [4], the authors present a method for investigating the genomic origins of sequenced reads and perform a large-scale analysis of public NGS samples. The key contribution of this paper is their method to report reads as being "unique-species-hit" or "multi-species-hit" to distinguish reads that could not be assigned a single source species of origin. To get a representative sample of the contamination landscape in public data, they downloaded and used their method to scan human RNA-seq datasets from ENCODE and Illumina. They determined around 5% of 432 RNA-seq samples were infected with mycoplasma. The paper included experimental data to support that ATAC-seq is more sensitive than RNA-Seq to report mycoplasma contamination. However, they did not address this question systematically by surveying public ATAC-seq libraries.

2.5 Current Tools and Services for Bacterial Detection

Over the years, there have been several tools developed to aid in the detection of contamination in NGS data. It is necessary for different tools to be developed as the technology producing NGS data is ever-evolving. New iterations of a procedure may change how or what the underlying data represents, which will make it incompatible with currently available tools.

2.5.1 OpenContami

OpenContami [13] is a web-based application for detecting microbial contaminants by authors from the same group as [4]. It offers several features to the research community:

1. User-friendly interface that allows users to upload their own data for analysis by the OpenContami pipeline.
2. The user can optionally open or close records to the public domain.
3. Results from users and public databases are continuously incorporated and used as a reference for future assessments.

However, there are several limitations to this service. First, they require the user to process their data to produce a BAM file that includes host-unmapped reads and then manually determine and input the number of host-mapped reads to the service. The processing pipeline itself, while extremely thorough and in-depth, is also a very computationally expensive task and could not be run in a reasonable amount of time locally on most computers. Thus, it demands a supercomputer to host the server. Most of the samples analyzed by this site are RNA-seq data. As of the time of this writing, only 15 ATAC-seq samples have been processed and are included.

2.5.2 Kraken 2

Kraken 2 [14] is a tool for assigning taxonomic labels to sequencing reads. It is an updated form of the original Kraken [15] with much lower memory usage and better performance. It is more similar to BLAST than other alignment tools because it is designed to classify reads from many species using a pre-built multi-specie database. The standard database most widely used with Kraken 2 includes the complete genomes for the bacterial, archaeal, and viral domains. The output of Kraken 2 is a report consisting of one line per input read, each with five fields:

1. One letter code indicating the sequence was either classified or not.
2. The sequence ID.
3. The classified taxonomy ID.
4. Length of the sequence in base-pairs.
5. Space-delimited list indicating the LCA (lowest common ancestor) mapping of each *k*-mer in the sequence.

Kraken 2 can convert these reports into sample-wide results for a human-readable summary of the taxonic makeup of individual samples. This summary includes information on the percentage of fragments covered by each taxon down to the species level [16]. Although Kraken 2 has made major improvements in memory, disk space, and speed usage, it still requires a relatively high memory footprint as well as a CPU-intensive preprocessing step to construct the required database.

Chapter 3: Methods

3.1 Data acquisition

ATAC-seq data was downloaded from the sequencing read archive (SRA) [1] using NCBI tools [17]. To compile a list of all currently available ATAC-seq projects, we used a table provided by the ChIP-Atlas project [18]. ChIP-Atlas is a data mining suite for chromatin immunoprecipitation sequencing (ChIP-seq) and DNase-seq data, of which ATAC-seq is a part of. Single-cell ATAC and other project types were excluded before downloading. To save storage space, only 100,000 reads were downloaded for each run to serve as a representative sample. Due to data formatting issues, we started from the 10,000th read [19]. Runs were discarded from further analysis if they did not contain at least 100,000 reads.

3.2 Alignment-based methods of contamination detection

For determining the sources of contamination within each sample, we used both Bowtie 2 [20] and Kraken 2 [14] to compare different approaches. Bowtie2 is commonly used for epigenetic sequencing analysis, while Kraken is preferentially used for metagenome analysis.

3.2.1 Using Bowtie 2

For Bowtie 2, whole genomes for several bacterial genera were downloaded using ncbi-genome-download and compiled into an index using bowtie2-build. Some of the genera we included were: *Acinetobacter, Alistipes, Anaerostipes, Clostridium, Mycoplasma*, and *Stenotrophomonas*. For each sample, we used the --un parameter to filter out reads mapping to the host species (hg38), vectors, and Escherichia coli, which is commonly used to synthesize proteins including the Tn5 enzyme used by ATAC-seq. See **Figure 1** for a flowchart diagram of the alignment filtering process. Additional Bowtie 2 parameters used were --local, -X 1000, --nomixed, --no-discordant, and --dovetail. We then recorded the percentage of fragments aligned to each bacterial index. If we found a sample had reads that could not be aligned to any species, we used SSAKE and NCBI BLAST to discover the origin species of those reads and added the genus to our Bowtie2 index collection for future scans.

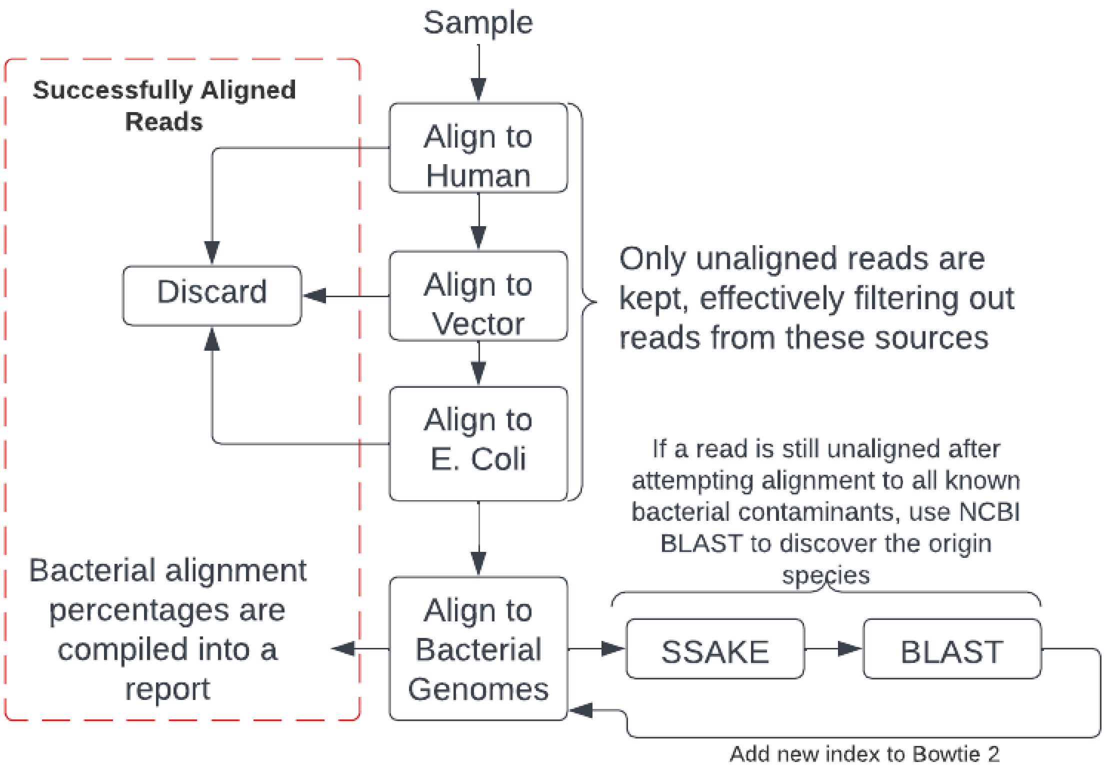
Sample
Successfully Aligned Reads
Align to Human
Discard
Align to Vector
Align to E. Coli
Only unaligned reads are kept, effectively filtering out reads from these sources
If a read is still unaligned after attempting alignment to all known bacterial contaminants, use NCBI BLAST to discover the origin species
Bacterial alignment percentages are compiled into a report
Align to Bacterial Genomes
SSAKE
BLAST
Add new index to Bowtie 2

Figure 1 - Bowtie 2 contamination pipeline flowchart

3.2.2 NCBI BLAST to Discover Novel Sources of Contamination

The Basic Alignment Search Tool (BLAST) is a bioinformatics resource for searching and aligning sequences hosted by the National Center for Biotechnology (NCBI). After inputting a source sequence, BLAST searches it against NCBI's database of nucleotide or protein sequences to determine the source organism or vector [21]. It then provides a detailed report on the statistical likelihood of possible sources for each input sequence. We used SSAKE [22] to assemble sequences for BLASTing.

The purpose of utilizing SSAKE and BLAST was to discover the source species of reads that could not be assigned using our current collection of Bowtie 2 indexes. We used the following pipeline to determine what species to add to our collection of genera to scan for:

1. Filter out reads from known sources (hg38, vectors, current list of bacterial sources).
2. Use SSAKE to reconstruct genomic sequences from the leftover reads of unknown origin.
3. BLAST the reconstructed sequence.
4. Inspect the report from BLAST for each sequence. If a sequence shows a statistically significant chance (E Value < 10^{-6}) of being from an unaccounted-for bacterial species, then we add the index for its genera to our list.

Species discovered using this method are *Staphylococcus Aureus, Ralstonia Solanacearum*, *Bos Mutus,* and *Cervus Elaphus*.

3.2.3 Using Kraken 2

To use Kraken 2, we downloaded and built the standard database, which includes the complete genomes for bacterial, archaeal, and viral domains, along with the human genome and a collection of known vector sequences. Kraken 2 was run on each sample and the final report was saved for further processing and analysis.

3.3 Our alignment-free, machine learning-based method

One active and interesting field of study is alignment-free analysis of NGS data. As we found when using Bowtie 2, aligning short-reads to a reference genome can be quite costly in terms of processing time and storage space, so finding ways to gain valuable information about sequencing data while skipping the alignment step would be a great way to improve the computational efficiency of our research.

For this report we devised a method to estimate the percentage of reads in a sample originating from the bacterial domain. The basis for our method is the fact that ATAC-seq data from human cells shows a particular pattern in the distribution of the length of DNA fragments. This is due to the presence of chromatin -DNA wrapping around nucleosome composed of histone proteins- in eukaryote but not in bacteria.

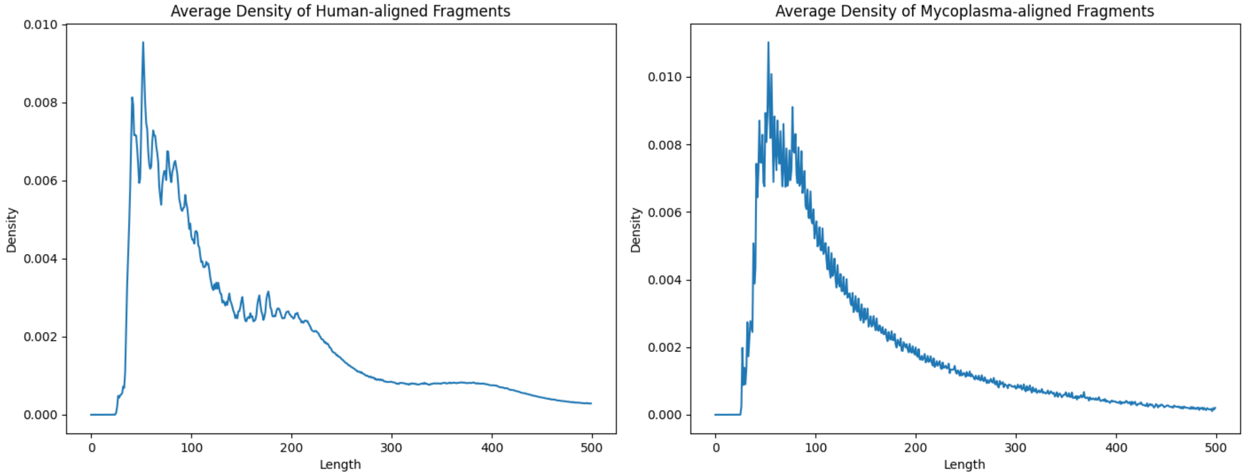

Figure 2 - ATAC-seq fragment length distributions

Figure 2 shows that ATAC-seq fragments mapped to human and mycoplasma are distinctive in fragment size distribution. On the left is the average distribution of fragment sizes from reads mapped to the human genome for all 20,000 samples. The graph on the right is the average distribution of fragment sizes from reads mapped to Mycoplasma.

Although the distributions are quite similar, there are two differences that we can use to distinguish them from each other. The first difference is the local maximum peak in the human plot at around 200bp (base pairs) and another at around 400bp. These peaks are caused by nucleosomes, which are structures made up of DNA coiling around a core of histones. Each coil is approximately 146bp long and connected by 80bp long linker DNA regions. In ATAC-seq, the hyperactive transposase Tn5 is more likely to slice the DNA somewhere within the linker DNA region, resulting in a peak every 200 base pairs [23]. On the other hand, bacteria do not have nucleosomes, so this pattern does not appear in its distribution.

The second difference is that the human fragment lengths have a distinct 10.5bp sine wave pattern. This is due to the natural twisting structure of the DNA causing base pairs facing away from the nucleosome to be more exposed and therefore more likely to be sliced. As mentioned previously, bacteria do not have nucleosomes so this phenomenon does not occur.

To make use of these distinctive differences, we can use our novel method of estimating the lengths of DNA fragments in paired FASTQ files and build a classification or regression machine learning model to detect contamination. This would give us a method of scanning samples for bacterial contamination without doing alignment, which would be much more efficient. The input to our model is constructed by first using our method to estimate all the fragment lengths from a sample and then applying the normalization procedure described in section 3.3.2, giving us a final array of 60 elements, which is then used as input to the machine learning model. See **Figure 3** for a flowchart diagram of the process. For the classification model, the output is an indication if the sample is predicted to be more than 20% contaminated. For our regression model, the output is a value between 0 and 100 that represents the percentage of reads that originated from bacterial sources. For training, we used the percentage reported under the bacterial domain from Kraken 2, as that is currently the most accurate tool available.

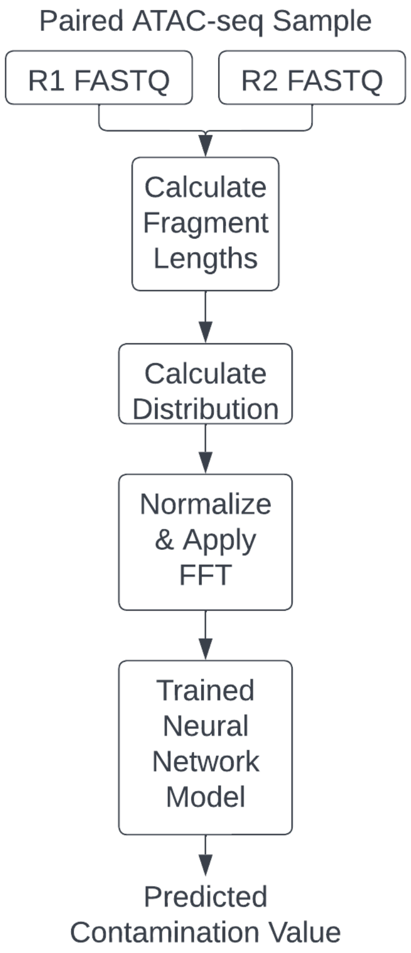
Paired ATAC-seq Sample
R1 FASTQ
R2 FASTQ
Calculate Fragment Lengths
Calculate Distribution
Normalize & Apply FFT
Trained Neural Network Model
Predicted Contamination Value

Figure 3 - Machine learning contamination pipeline flowchart

3.3.1 Estimating fragment length distribution via pattern matching

In order to understand how to estimate fragment lengths from ATAC-seq data without aligning to a reference genome, we must first introduce the structure in which the data is stored. ATAC-seq sequencing data is commonly stored in the pair-end FASTQ format. This format was invented in the early 2000's at the Wellcome Trust Sanger Institute by Jim Mullikin [24]. Each FASTQ file is made up of many records called a "read". Each read contains 4 lines of information:

1. A sequence identifier with information pertaining to the run or sample. This is non-standard and provides different information depending on how the FASTQ file was generated.
2. The sequence (a string of A, C, T, and G in ASCII).

3. A separator line, which starts with a '+' character. This line may optionally provide more information about the sequence or comments.
4. The base call quality scores. These are encoded characters paired with the sequence that show the quality of each base-pair in the sequence.

Additionally, most ATAC-seq samples are "paired-end", meaning that nucleotide sequences were read from both ends of the DNA fragment and stored in paired FASTQ files. Read 1 starts at the 5' end of the DNA strand and extends towards the 3' end along the forward DNA strand. Read 2 starts at the 3' end of the DNA strand and extends towards the 5' end along the reverse DNA strand [25]. These reads are stored in separate FASTQ files where the filename matches either *Database_RunID_[12].fastq* following the sequence read archive format or *SampleName_SampleNumber_Lane_R[12]_FlowCellIndex.fastq.gz* following the Illumina-style format [26]. By finding the overlap between the paired reads we are able to estimate the original length of the DNA fragment from which that read came from **(Figure 4**).

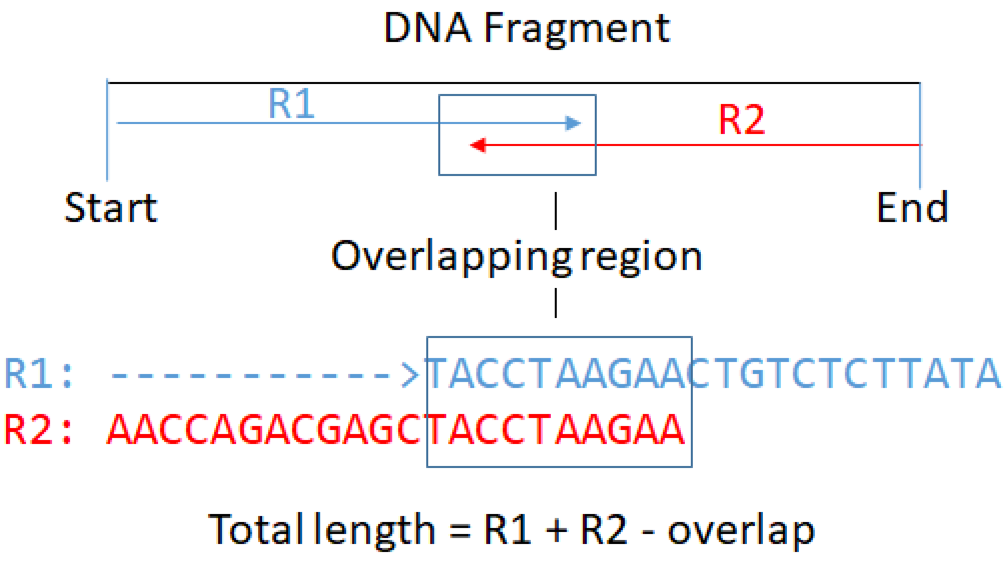
DNA Fragment
R1
R2
Start
End
Overlapping region
R1: ------------>TACCTAAGAACTGTCTCTTATA
R2: AACCAGACGAGCTACCTAAGAA
Total length = R1 + R2 - overlap

Figure 4 - Illustration of estimation of DNA fragment length

Figure 4 is an illustration of how the estimation algorithm works for an example DNA fragment. R1 has been shifted over to show the overlapping sequence pattern. To calculate the total length of the fragment in base pairs, add the lengths of R1 and R2 together and subtract the length of the overlapping region. In this example, |R1| = |R2| = 22 and |overlap| = 10, so the length of the fragment is 34. The sequence for R2 has already been translated to its reverse complement for this example.

In brief, our algorithm is based on simple pattern searching except we report a match when we are able to match until the end of the text, not until the end of the pattern like in most pattern searching problems. We also require a minimum overlap length of at least 15 base pairs to prevent falsely reporting a length in the rare case where sequences overlap purely by chance. A description of the naïve algorithm follows:

```
Input:
R1 = read 1, a string of the letters ACTG representing the sequence
R2 = read 2
L = length(R1)
Frag_length = 0
i = 0
shift=0
found=false
while (shift < L – 15 & !found)
{
        while (R1[i] == R2[i + shift] & i + shift < L)
        {
                i = i+1
        }
        if (i + shift == L)
        {
                found = true
                break
        }
        shift = shift+1
if (found)
        Frag_length = L + shift
```

```
20. return Frag_length
```

After applying this algorithm, we can obtain a fairly accurate estimate size of the fragments where the length is greater than L but less than 2*L-15. Of course, this algorithm can be improved by utilizing known pattern searching techniques such as the KMP method.

3.3.2 Normalizing Distribution Frequency via FFT

As stated previously, our algorithm can only estimate the size of the fragments whose length are less than 2*L-15 but greater than or equal to L. The issue with this is that ATAC-seq libraries have different read lengths due to differences in protocol or sequencing machine used. So for example, if one library has a read length of L=51 for all runs, then we can only estimate the lengths for fragments in the range [51, 86]. Whereas for a different library with a read length of L=76, we can make estimates in the range [76,136]. In order to increase similarity and comparability between libraries, after calculating the distribution *I*, we shift each element in the array to the left L times so that the minimum length for each library is always 0. We then remove reads greater than 120bp and apply a Fourier transform to *I*.

The rationale behind using this transformation is that the shape of the distribution is more important for the model to learn and not the actual values. Specifically, we need the machine learning model to recognize the 10.5bp periodicity sine wave pattern due to the twisting structure of DNA, see **Figure 8**. After applying the shift, the sine wave will be out of phase when comparing libraries with different read lengths, which could make recognizing that pattern more difficult for a neural network. Applying the Fourier transform and taking only the real coefficients effectively removes the phase information and calculates the magnitude of the 10.5bp pattern for us, which is less pronounced in bacteria, see **Figure 2**. Also recall earlier we mentioned that the shape of the distribution is different for bacterial reads due to nucleosomes. See **Figure 5** for an example fragment length distribution before and after this transformation. Despite this library having a small read length, we can still note the higher energy around the 11th and 12th coefficients in the frequency domain graph, which corresponds to the magnitude of the 10.5bp pattern.

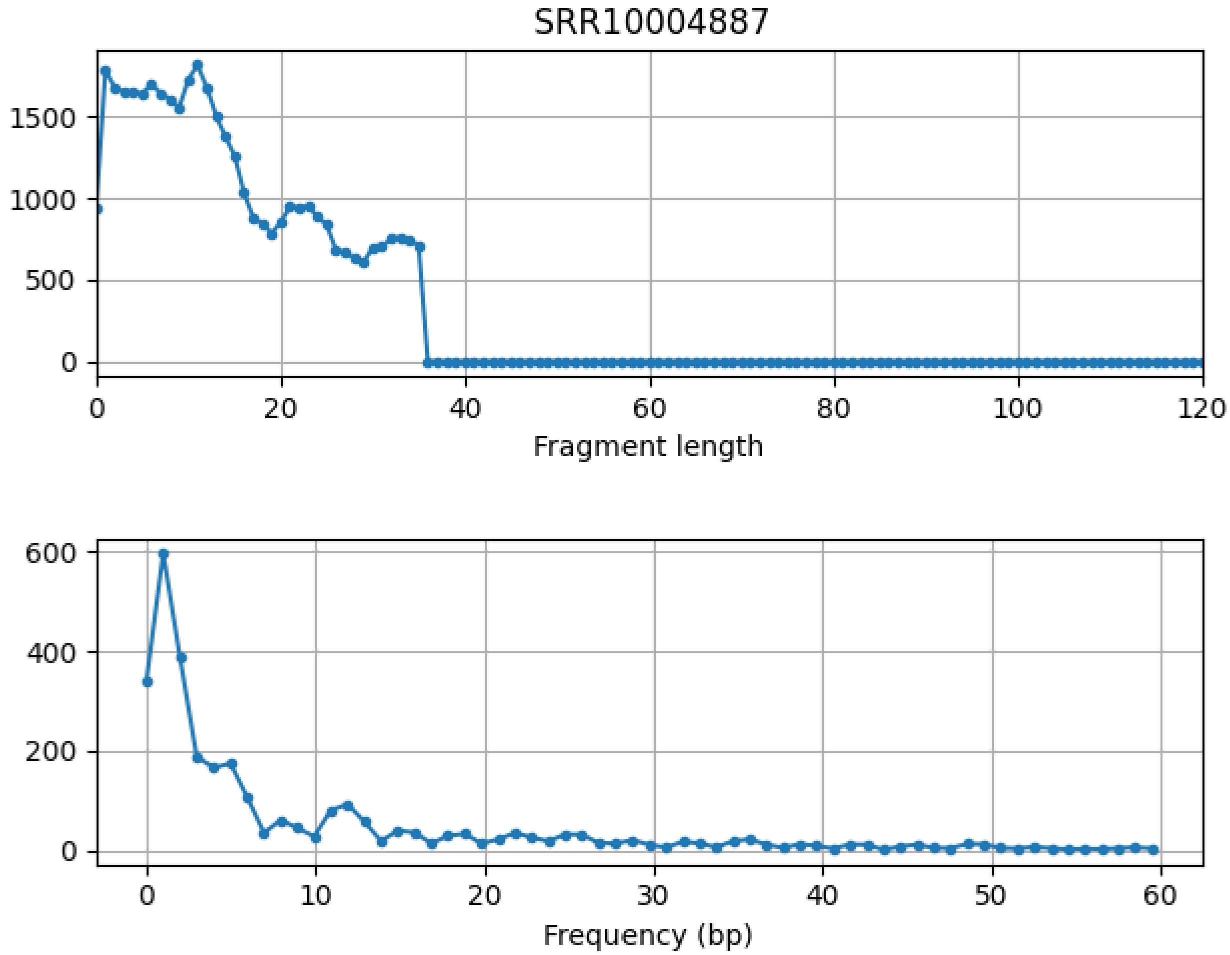

Figure 5 - Example fragment length distribution

3.3.4 Deep learning architecture

In order to develop a simple neural netwsork model that can estimate the overall level of contamination in future ATAC-seq samples, I used the Keras [27] and sklearn [28] python packages. I chose these tools as they provide convenient high-level interfaces to a multitude of common machine learning and artificial intelligence procedures.

I experimented with several different architectures, activation functions, dropout rate, losses, and optimizers. I found that large, wide architectures would quickly overfit to the training data and would not perform well on the testing data for performance assessment. This is possibly due to having a rather limited number of samples to use for training, especially contaminated samples. To help the neural network learn the more abstract features of the data and prevent it from simply memorizing the input data, I used a smaller neural network.

The architecture I eventually settled on is the following:

1) **Input layer** with a shape of 61x1. It has only 61 features because after limiting the range of fragment lengths to 120 and computing the discrete Fourier transform the results include both positive and negative-frequency terms. In our case, the negative-frequency terms are just inverted copies of the positive-frequency terms and provide no information, so are discarded.
2) **Dense layer** with 64 nodes activated via the standard ReLU function: max(x, 0).
3) **4 Dense layers** with 128 nodes each also using the ReLU function.
4) **Dense layer** with 64 nodes, this time using a sigmoid activation function: 1 / (1 + exp(-x)).
5) **Output layer** with a single node.

3.3.5 Training & Validation: Regression

For training, I reserved 30% of the samples for testing and 70% for training. I used the Adam optimizer algorithm, [29] which is a computationally efficient extension to stochastic gradient descent that works well with noisy data [30]. The rationale is that the data we are working with was generated in real-world lab conditions using differing protocols. To compute the loss during training and testing, I used the mean squared error loss function. The mean squared error loss function computes the mean of squares of errors between labels and predictions using the formula $MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2$ where Y represents the true values and Ŷ represents the predicted values from the neural network.

3.3.6 Training & Validation: Classification

For training the classification model, we defined all samples with >20% of reads originating from bacterial sources according to Kraken 2 as being contaminated. We used a similar model as in the previous section, except using binary cross entropy as our loss function and binary accuracy as a metric for monitoring. After training several of these models, we saw that although the accuracy was reported to be greater than 95%, it would sometimes completely misclassify all the contaminated samples. This was because our dataset was heavily imbalanced and we needed to find a method to mitigate this issue.

To force our model to be more sensitive to contaminated samples, we used a method known as naïve random over-sampling implemented in the imbalanced-learn python library [31]. This method generates new samples by randomly sampling from the currently available samples. Using this, we were able to generate a new dataset where the distribution between the contaminated and non-contaminated samples was the same. This greatly increased the model's accuracy in identifying the contaminated samples, while introducing some false positives, which we considered to be more acceptable than false negatives for our specific use case.

All models were saved using Kera's model.save function and the correlation between the predicted contamination values from the model and actual values were inspected using Matplotlib, see **Figure 9** [32].

Chapter 4: Results

In this section we summarize our findings on the prevalence of contamination found in the SRA database and the accuracy of our method for estimating fragment lengths from raw ATAC-seq FASTQ files, as well as the results from our machine learning models.

4.1 Prevalence of Contamination in SRA Database

After downloading each run stored in the SRA database and performing a systematic scan of bacterial contaminants via the 2 different alignment-based methods, we found that between 3-5% of the 20,000 samples have over 20% of their reads possibly originating from bacterial sources. Kraken 2 reports 4.8% of the samples pass this contamination threshold, while Bowtie 2 reports 3.5%. **Figure 7** shows the top 10 sources of contamination for each tool.

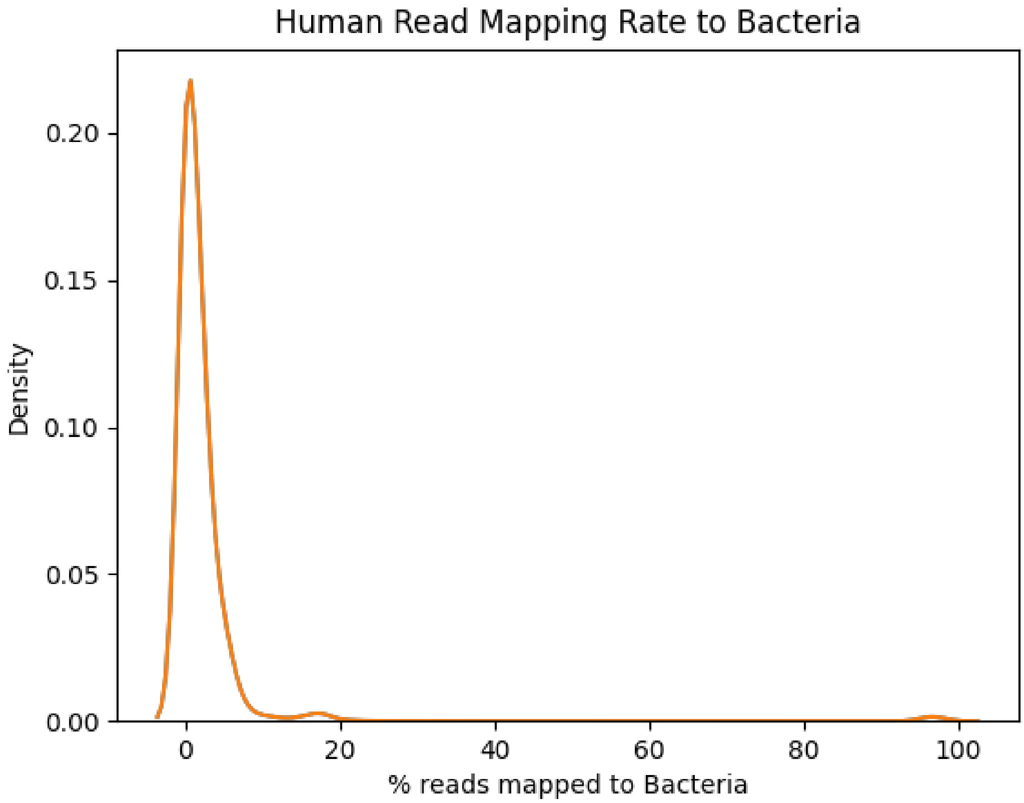

Figure 6 - Percentage of Human Reads That Also Map to Bacteria

]

Figure 6 shows why we chose a threshold of 20% to mark a sample as contaminated. Due to the nature of genetics, there is expected to be some overlap in the DNA sequences of bacteria and humans. To determine the amount of overlap, we first filtered all the samples to only reads we could map to human, and then scanned those reads for bacterial contamination using Kraken 2. The majority of samples showed 0% of their reads as possibly originating from the bacterial domain, but there were some outliers that still reported up to 20%.

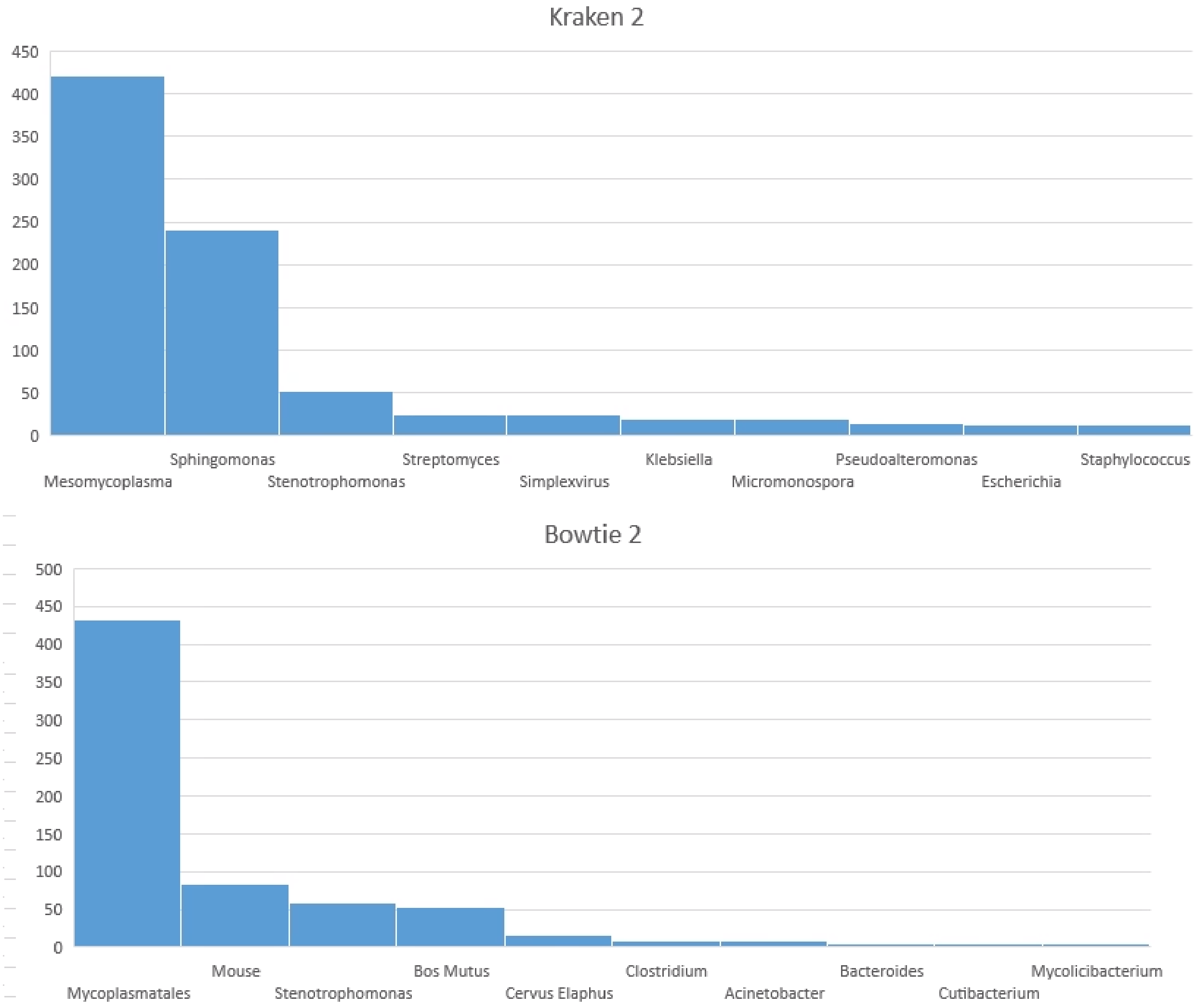

Figure 7 - Top sources of contamination

As expected, the most common source of contamination originates from the order *Mycoplasmatales*, of which *Mesomycoplasma* is a genus of. On the other hand, we were surprised that out of the top sources, the only other one in common between the two tools were *Stenotrophomonas*. A possible explanation for this is that Kraken 2 has a much larger and more complete database of bacterial genera, as well as the mutual exclusion feature mentioned previously.

4.2 Performance of Fragment Length Estimation Algorithm

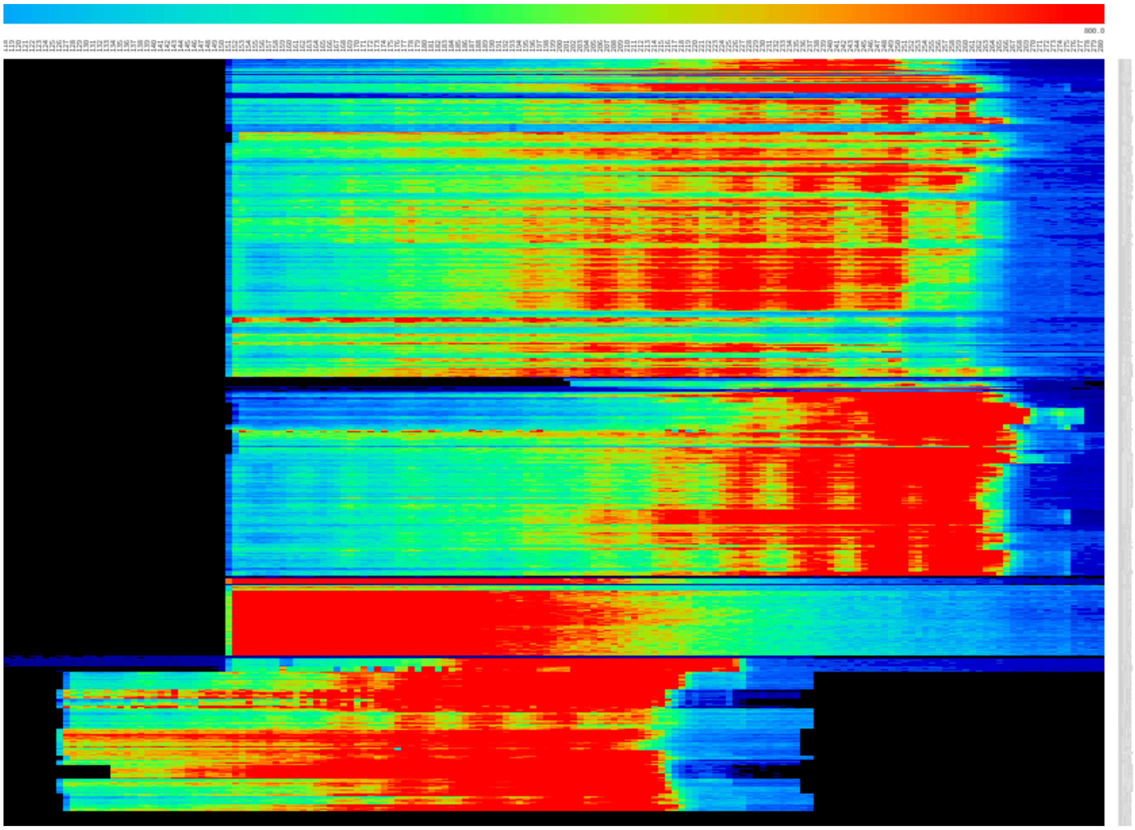

Figure 8 - Hierarchical clustering heatmap of estimated fragment length distributions

Figure 8 is a hierarchical clustered heatmap showing the fragment length distribution for two ATAC-seq studies. Each row is a separate sample, and each column is a fragment length. The color intensity at each point is represents how many fragments from a sample has a specific length. The minimum length in this figure is 120 and the maximum is 280.

As you can see, a regular 10.5bp periodicity is visible in the fragment size distributions. This pattern is caused by the pitch of the DNA strands exposing certain regions more readily to the hyperactive transposase Tn5 while wrapping around a nucleosome [23]. Also visible in **Figure 8** is the difference in the minimum and maximum estimable fragment lengths for two different studies.

4.3 Performance of Neural Network Model

4.3.1 Regression Accuracy

After our experiments with different architectures, loss functions, and sampling methods, the best mean squared error we could achieve was *MSE = 30.*

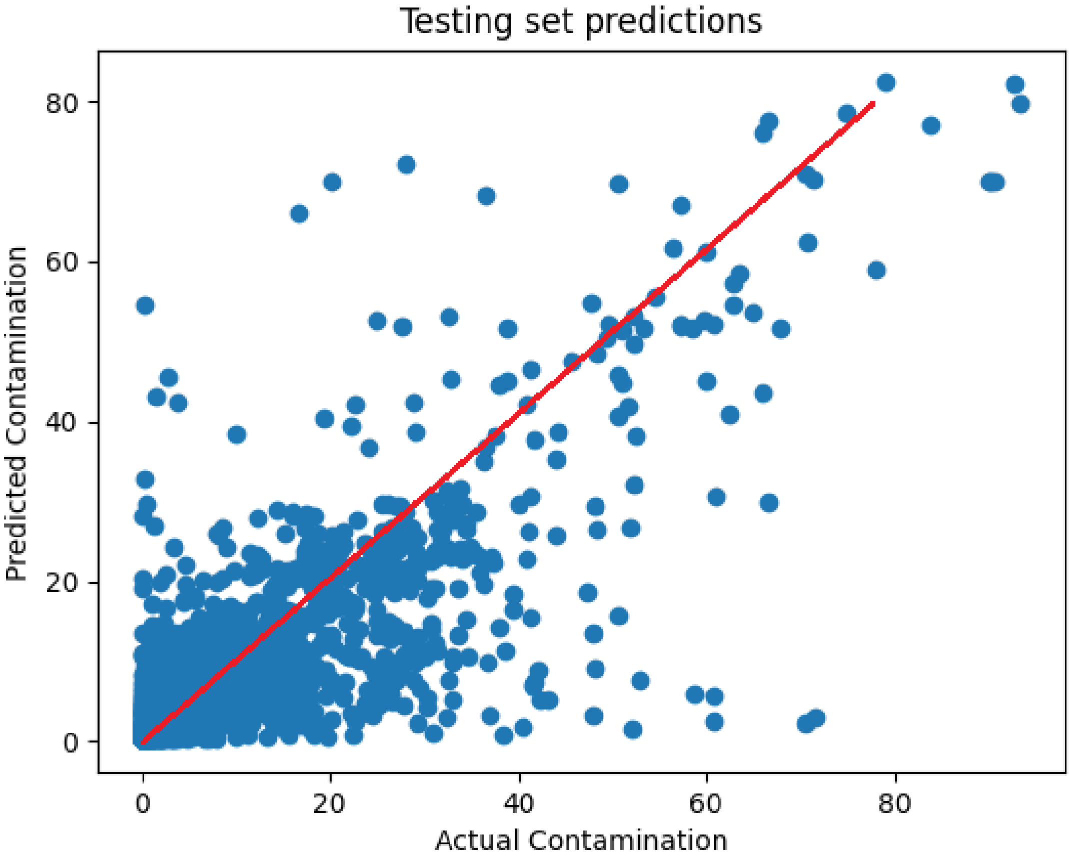

Figure 9 - Scatter plot of actual vs predicted contaminated values

Figure 9 is a scatter plot of the actual contaminated values on the x-axis and the predicted values on the y-axis. The red line shows the hypothetical ideal regression line (a perfect one to one correlation between the two values), the further away a point is from this line, the greater the error.

Due to the relatively low accuracy of our regression model, but signs that there is some correlation, we hypothesized that accuracy and usability may be improved if we restated our goal as a classification problem. For example, labelling a sample with greater than 20% of its reads as "contaminated" and training a model to classify a sample as either contaminated or not may perform better than trying to predict exact percentages.

4.3.2 Classification Accuracy

Preliminary classification results looked much better than our attempt at regression, with a reported accuracy of 95%. However, this number was misleading, as it was simply labelling all samples as "not contaminated", which made up the majority of the dataset. To teach the model to be more sensitive to contamination we used naïve random oversampling to handle the data imbalance problem. This introduced a few false positives, but greatly improved its ability to identify contamination, as seen in the confusion matrix shown in **Figure 10**. The result is a 97.07% overall accuracy in classifying the samples, see **Table 1**.

Table 1 - Classification model statistics

	precision	recall	f1-score
FALSE	0.98	0.99	0.98
TRUE	0.79	0.68	0.73
accuracy			0.97
macro avg	0.88	0.83	0.86
weighted avg	0.97	0.97	0.97

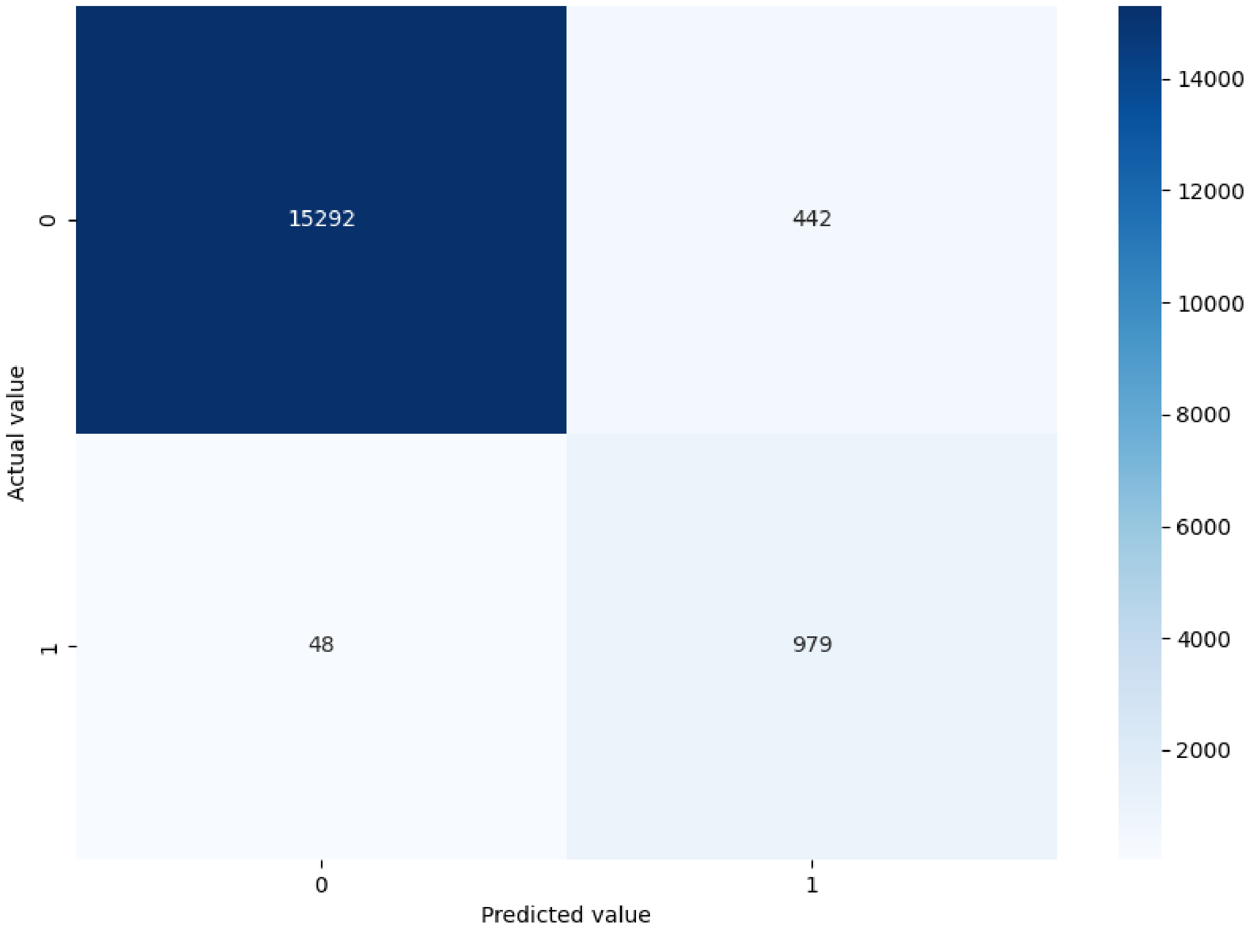

Figure 10 - Confusion matrix

4.4 Speed and Memory Usage

To measure the performance of each of the approaches, we used the GNU time utility. We ran each tool 3 times using the same samples for each and calculated the averages for each tool.

Table 2 - Single sample performance statistics

Tool	User time	System time	Elapsed "Wall clock" time	Maximum memory usage (kb)
Bowtie 2	29.58	2.86	00:32.9	3,777,493
Kraken 2	0.67	2.24	00:03.0	39,552,625
our_method	19.30	7.03	00:14.5	454,269

Table 2 shows the time and memory results from running each of the three tools on a single ATAC-seq sample, with no parallelization or multi-threading enabled. We found that Kraken 2

was the fastest by an order of magnitude and used about 40GB of memory. Bowtie 2 was much slower and used 3.6GB. Our machine learning method was slower than Kraken 2, but faster than Bowtie 2 while using only 0.4GB. Due to the lower memory requirements of our tool, we hypothesized that it may be more conducive to parallelization than the others.

Table 3 - 200 Sample Multi-Threaded Performance

Tool	User time	System time	Elapsed "Wall clock" time	Maximum memory size (kb)
Bowtie 2	26401.20	8271.75	11:09.3	16,805,203
Kraken 2	339.97	556.92	00:45.1	53,591,052
our_method	4343.20	60.83	01:36.8	419,117

Table 3 shows the results from running each tool on 200 samples in parallel. To efficiently utilize available cores and file I/O, we tuned the number of threads and parallel instances of each program. For Bowtie 2, we used 8 threads per process and ran 8 alignment jobs in parallel. Additionally, we used the --mm option to load the indexes into shared memory that can be used by each instance, reducing loading times and decreasing overall memory usage. For Kraken 2, we used 4 threads and ran 16 processes in parallel. Kraken 2 also provides a --memory-mapping option, which we utilized. Since our method for estimating fragment lengths is only single-threaded, we ran 64 of them in parallel, compiled an array of the completed results, and ran the prediction model only once on the final array. This idea proved quite effective at optimizing parallel performance, as our method shows a 29X speedup when run in parallel, compared to a 13X speedup for Kraken 2 or a 9.5X speedup for Bowtie 2. Speedup is defined as the ratio of serial execution time to parallel execution time [33].

All tests were performed on a computer with 2 Xeon Gold 6242 CPUs @ 2.8GHz and 512GB of RAM, 64 cores in total.

Chapter 5: Conclusion & Discussion

5.1 Discussion

In this research, we perform a systematic scan on human ATAC-seq samples publicly available from the SRA database and find around 5% of the samples contain traces of bacterial contamination, mainly originating from mycoplasma. This implies the proportion and source of contamination in public ATAC-seq data parallels previous reports pertaining to public RNA-seq samples [2], [12]. Since samples are uploaded as-is to SRA, the task of quality control falls upon the researchers who wish to utilize this vast resource of experimental data. This can lead to the pollution of public databases and waste the time of future researchers who draw false results from the contaminated data [34].

To explore the current state of the available tools for quality control, we compare several common alignment-based methods of contamination detection, namely Bowtie 2 and Kraken 2. In the interest of researching alignment-free methods of analysis of genomic sequences, we train a neural network to detect contaminated samples based on the sample's distribution of DNA fragment lengths. Our model achieves an accuracy of 97% and uses an order of magnitude less memory than current state of the art alignment methods.

Upon completion of the Bowtie 2 pipeline, we have implemented it as a key quality control step in our lab. It has been used to scan over 1,000 samples sent to us by collaborators before we begin further data analysis.

The code for this project is available on GitHub: https://github.com/Hu-sLab/Bacterial-Contamination-in-ATAC-seq

5.2 Limitations of This Work

The main limitation of this work is that we purposefully limited our scope to human ATAC-seq samples. The contamination landscape may be quite varied for different organisms. For example, cells collected from experimental mice without culture would be less likely exposed to mycoplasma. Future work could include data from single-cell ATAC-seq experiments, which is an even newer technology that requires some specialized processing.

A limitation of our machine learning model is that our dataset is rather small and unbalanced for efficient and accurate training, given that there are far less contaminated samples than normal samples. Future work may look into utilizing more sophisticated forms of synthetic data generation or data augmentation techniques to help mitigate this issue. Our fragment length estimation approach may also be able to be combined with other alignment-free analyses methods such as *k*-mer frequency.

References

[1] R. Leinonen, H. Sugawara and M. Shumway, "The Sequencing Read Archive," *Nucleic Acids Research,* vol. 39, no. 1, pp. D19-D21, 2011.

[2] M. Strong, G. Xu, L. Morici, S. Bon-Durant, M. Baddoo, Z. Lin, C. Fewell, C. Taylor and E. Flemington, "Microbial Contamination in Next Generation Sequencing: Implications for Sequence-Based Analysis of Clinical Samples," *PLOS Pathogens,* vol. 10, no. 11, 2014.

[3] C. Miller, H. Kassem, S. Pepper, Y. Hey, T. Ward and G. Margison, "Mycoplasma infection significantly alters microarray gene expression profiles," *Microarray Technologies,* vol. 35, pp. 812-814, 2003.

[4] S.-J. Park, S. Onizuka, M. Seki, Y. Suzuki, T. Iwata and K. Nakai, "A systematic sequencing-based approach for microbial contaminant detection and functional inference," *BMC Biology,* 2019.

[5] C. Wilson, R. Nowell and T. Barraclough, "Cross-Contamination Explains “Inter and Intraspecific Horizontal Genetic Transfers” between Asexual Bdelloid Rotifers," *Current Biology,* vol. 28, no. 15, pp. 2436-2444.e14, 2018.

[6] CDC, "What is Epigenetics?," CDC, 2022. [Online]. Available: https://www.cdc.gov/genomics/disease/epigenetics.htm. [Accessed 1 7 2022].

[7] S. Klemm, Z. Shipony and W. Greenleaf, "Chromatin accessibility and the regulatory epigenome," *Epigenetics,* vol. 20, pp. 207-220, 2019.

[8] J. D. Buenrostro, B. Wu, C. Y. Howard and W. J. Greenleaf, "ATAC-seq: A Method for Assaying Chromatin Accessibility Genome-Wide," *Curr. Protoc. Mol. Biol.,* pp. 109:21.29.1-21.29.9. doi: 10.1002/0471142727.mb2129s109, 2015.

[9] M. Tsompana and M. Buck, "Chromatin accessibility: a window into the genome," *Epigenetics & Chromatin,* p. 7:33, 2014.

[10] F. Peres and D. Riano-Pachon, "ContFree-NGS: Removing Reads from Contaminating Organisms in Next Generation Sequencing Data," *Brazilian Symposium on Bioinformatics,* vol. 13063, pp. 65-68, 2021.

[11] M. Sangiovanni, I. Granata, A. Thind and M. Guarracino, "From trash to treasure: detecting unexpected contamination in unmapped NGS data," *BMC Bioinformatics,* vol. 20, p. 168, 2019.

[12] A. Olarerin-George and J. Hogenesch, "Assessing the prevalence of mycoplasma contamination in cell cultre via a survey of NCBI's RNA-seq archive," *Nucleic Acids Research,* vol. 43, no. 5, pp. 2535-2542, 2015.

[13] S.-J. Park and K. Nakai, "OpenContami: a web-based application for detecting microbial contaminants in next-generation sequencing data," *Bioinformatics,* pp. 3021-3022, 2021.

[14] D. E. Wood, J. Lu and B. Langmead, "Improved metagenomic analysis with Kraken 2," *Genome Biology,* vol. 20, no. 1, 2019.

[15] D. E. Wood and S. L. Salzberg, "Kraken: ultrafast metagenomic sequence classification using exact alignments," *Genome Biology,* 2014.

[16] J. Lu and B. Langmead, "Kraken 2 Manual," December 2020. [Online]. Available: https://github.com/DerrickWood/kraken2/wiki/Manual. [Accessed July 2022].

[17] National Center for Biotechnology Information, "Entrez Programming Utilities Help," National Center for Biotechnology Information, 2010. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK25501/. [Accessed 7 July 2022].

[18] Z. Zou, T. Ohta, F. Miura and S. Oki, "ChIP-Atlas 2021 update: a data-mining suite for exploring epigenomic landscapes by fully integrating ChIP-seq, ATAC-seq and Bisulfite-seq data," *Nucleic Acids Research,* 2022.

[19] R. Edwards, "fastq-dump," Edwards' Lab, 29 December 2015. [Online]. Available: https://edwards.flinders.edu.au/fastq-dump/. [Accessed July 2022].

[20] B. Langmead and S. Salzberg, "Fast gapped-read alignment with Bowtie 2," *Nature Methods,* vol. 9, pp. 357-359, 2012.

[21] G. Boratyn, C. Camacho, P. Cooper, G. Coulouris, A. Fong, N. Ma, T. Madden, W. Matten, S. McGinnis, Y. Merezhuk, Y. Raytselis, E. Sayers, T. Tao, J. Ye and I. Zaretskaya, "BLAST: a more efficient report with usability improvements," *Nucleic Acids Research,* vol. 41, no. W1, pp. W29-W33, 2013.

[22] R. Warren, G. Sutton, S. Jones and R. Holt, "Assembling millions of short DNA sequences using SSAKE," *Bioinformatics,* vol. 23, no. 4, pp. 500-501, 2007.

[23] J. Buenrostro, P. Giresi, L. Zaba, H. Chang and W. Greenleaf, "Transposition of native chromatin for multimodal regulatory analysis and personal epigenomics," *Nat Methods,* pp. 1213-1218, 2013.

[24] P. Cock, C. Fields, N. Goto, M. Heuer and P. Rice, "The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants," *Nucleic Acids Research,* vol. 38, pp. 1767-1771, 2010.

[25] The Sequencing Center, "What are paired-end reads?," The Sequencing Center. [Online]. [Accessed 3 7 2022].

[26] Illumina, "FASTQ File Upload Requirements," Illumina. [Online]. [Accessed 3 7 2022].

[27] Chollet, Francois and et. al, "Keras," 2015. [Online]. Available: https://keras.io.

[28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research,* vol. 12, pp. 2825-2830, 2011.

[29] D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *Published as conference paper at ICLR,* 2015.

[30] J. Brownlee, "Gentle Introduction to the Adam Optimization Algorithm for Deep Learning," machinelearningmastery, 13 January 2021. [Online]. Available: https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/. [Accessed 5 July 2022].

[31] G. Lemaitre, F. Nogueira and C. Aridas, "Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning," *Journal of Machine Learning Research,* vol. 18, no. 17, pp. 1-5, 2017.

[32] J. D. Hunter, "Matplotlib: A 2D Graphics Environment," *Computing in Science & Engineering,* vol. 9, no. 3, pp. 90-95, 2007.

[33] L. Akyil, C. Breshears, M. Corden, J. Fedorova, P. Fischer, H. Gabb, V. Gromova, J. Hoeflinger, R. Hubbard, A. Kukanov, K. O'Leary, D. Ott, E. Palmer, A. Pegushin, P. Petersen, T. Rosenquist, A. Tersteeg, V. Tsymbal, M. Voss and T. Zipplies, "Predicting and Measuring Parallel Performance," 2011. [Online]. Available: https://www.intel.com/content/dam/develop/external/us/en/documents/1-1-appthr-predicting-and-measuring-parallel-performance-165587.pdf.

[34] NCBI, "Contamination in Sequence Databases," NCBI, 15 July 2016. [Online]. Available: https://www.ncbi.nlm.nih.gov/tools/vecscreen/contam/. [Accessed September 2022].