Stockholms
universitet

A Bayesian Approach to Clustering

Correcting Errors in DNA barcode reads

Nik Tavakolian

Masteruppsats i matematisk statistik
Master Thesis in Mathematical Statistics

Masteruppsats 2021:11
Matematisk statistik
Juni 2021

www.math.su.se

Matematisk statistik
Matematiska institutionen
Stockholms universitet
106 91 Stockholm

Mathematical Statistics
Stockholm University
Master Thesis **2021:11**
http://www.math.su.se

A Bayesian Approach to Clustering
Correcting Errors in DNA barcode reads

Nik Tavakolian*

June 2021

Abstract

DNA barcodes are short DNA sequences introduced into a population to track the relative frequencies of lineages over time. These barcode sequences are unknown to the human observer upon insertion and must be identified using next-generation sequencing technology. This process is error prone and results in a large number of error sequences. To estimate the relative frequencies of the barcodes accurately these errors must be corrected for. This error correction task can be posed as a clustering problem where the goal is to group similar sequences together. Existing methods for this task have used the observed frequency of the sequences but have disregarded the per nucleotide error rate in the clustering process. Without an accurate estimate of this error rate the distribution of error sequences cannot be inferred, limiting the error correction accuracy of these methods. Furthermore, these methods have delegated the task of parameter selection to the user, leaving room for user errors resulting from unsuitable parameter choices. In this work we set out to develop a clustering procedure that addresses these shortcoming. We estimate the per nucleotide error rate and devise a Bayesian hypothesis test for distinguishing between true barcodes and error sequences. The proposed method considers all nearby sequences before clustering a given sequence and achieves higher accuracy than the current state-of-the-art method on simulated datasets.

*Postal address: Mathematical Statistics, Stockholm University, SE-106 91, Sweden. E-mail: nik.tavakolian@gmail.com. Supervisor: Chun-Biu Li.

Contents

Notation 1

1 Introduction 2

2 Mathematical Description 4

3 Method 9

3.1 The k-mer index 9

3.2 Clustering Procedure 13

3.3 Bayesian Hypothesis Test 14

3.3.1 Marginal Likelihood of Model M_1 15

3.3.2 Marginal Likelihood of Model M_2 17

3.3.3 Read Classification using Bayes Factor 18

3.4 Parameter Selection and Optimization 18

3.4.1 Determining ϵ 18

3.4.2 Determining k 19

3.4.3 Determining ρ 20

3.4.4 Performance Optimization 21

4 Results 23

4.1 Simulated Datasets 23

4.2 Evaluation 24

5 Discussion 27

A Appendix 30

Bibliography 31

Notation

l Barcode length including only the random nucleotide positions, i.e excluding constant regions that are the same across all barcodes.

k Length of the substrings used to construct the k-mer index

p Number of partitions ($p = l/k$)

ϵ Maximum hamming distance considered for merging two sequences

ρ Estimated per nucleotide error rate

S_c Sequence under consideration - to be classified as either a true barcode or an error sequence originating from S_b

S_b Closest putative barcode in the ϵ-neighborhood of S_c

d Hamming distance between sequences S_c and S_b

τ Merging distance threshold for frequency 1 reads

f Frequency threshold for classifying reads as error sequences

1 Introduction

DNA barcodes are short DNA sequences that are introduced into a population to identify individuals and their offspring. These barcodes are passed on from generation to generation and can be used to track the relative frequencies of different lineages over time. This technology is useful for analyzing the evolutionary dynamics of a population. For example, it has been used to infer the presence of mutations in populations of *Saccharomyces cerevisiae* (baker's yeast) and to track the progression of breast cancer in humans [5, 6].

In general, the barcodes are unknown random DNA sequences. Once established in a population the barcodes are copied using Polymerase Chain Reaction (PCR) in a process called PCR amplification. The PCR amplification process generates millions of copies of each DNA barcode. This is done to facilitate the next step of the process where next-generation sequencing is used to identify the DNA sequences of the barcodes. The number of times a particular DNA sequence was read by the sequencer gives us information about the frequency of the corresponding barcode in the population. If we compare the number of times each distinct DNA sequence was found by the sequencer we obtain an estimate of the relative frequency of each barcode in the population. However, this ignores the fact that the sequencing process is error prone and assumes that each identified sequence corresponds to a barcode in the population. Both PCR amplification and sequencing can introduce errors in the identification of the barcodes, typically in the form of substitution errors, whereby one or more nucleotides in a barcode are exchanged for different nucleotides. To correctly determine the relative frequencies of the barcodes these errors must be identified and corrected for. Indel errors that change the length of the barcodes by inserting and deleting nucleotides occur at a much lower rate and will not be accounted for in our error correction scheme [9]. Consequently, sequences of different lengths will be processed separately.

Define an error read as a sequencing read that introduced one or more substitution errors in the identification of a barcode. The original barcode is the source barcode of the error read. The error correction task can be viewed as a clustering problem where we want to group similar reads together. All error reads that have the same source barcode should belong to the same group together with their source barcode.

The main challenge of this task is that the number of unique DNA sequences found by the sequencer (unique reads) can be in the millions and the number of barcodes can be in the hundreds of thousands. Since clustering involves grouping similar items together a standard approach is to compute all pairwise distances between the items, before applying some clustering scheme. However, with millions of unique reads this approach is too computationally expensive in our case. In addition, the number of barcodes is unknown beforehand, making the task more difficult since the cluster count cannot be used as a guide to find correct clusters.

Bartender and Starcode are examples of existing methods for the specific task of clustering barcode reads that account for the distances between reads and the read frequencies [10, 11]. To make the task of finding sequence distances computationally tractable these methods use various prioritisation schemes to avoid computing all pairwise distances.

Nevertheless, existing methods do not account for the error rates associated with PCR amplification and sequencing. These are important factors to consider, allowing us to estimate the probability distribution of the error reads. This distribution can help us to accurately determine whether a given read is a barcode or an error read. It also enables a data-driven approach for automatic parameter selection which eliminates the possibility of user errors arising from inappropriate choices of parameters.

Here we propose a new method for the task of clustering barcode reads which seeks to improve accuracy by addressing the aforementioned shortcomings of existing methods. Our approach is based on the idea of partitioning the reads into non-overlapping substrings. These substrings are used as keys in an indexing scheme to efficiently find all reads that are similar to a given read. We use a Bayesian hypothesis test to accurately determine if a given read is an error read or a barcode. The main achievement of our approach is that it offers a substantial improvement in error correction accuracy over previous methods.

The thesis will be organized as follows. Section 2 provides a mathematical description of the data and presents the main challenges associated with the clustering problem. In section 3 we provide a detailed description of our method. In section 4 we evaluate the method on simulated datasets with comparisons to Bartender, the current state-of-the-art method. We discuss the results and the significance of our new approach in section 5 and conclude our findings.

2 Mathematical Description

In this section we will discuss the main challenges associated in clustering barcode reads. We will start by analysing the proximity of the barcodes in sequence space in the absence of errors. Then we will see how error reads from one barcode can get close to another barcode in some cases, posing a challenge when clustering the reads. Finally, we will discuss the practical challenges of the clustering problem.

Throughout this text we will assume that each row in the input dataset contains a unique read and its observed frequency in the population. The observed frequency is simply the number of times the sequence was read by the sequencer.

To analyse the proximity of the barcodes and to perform clustering of the reads we need to start by defining a sensible distance between the DNA sequences. We will process sequences of different lengths separately and consider the cases when error reads arise as a result of substitutions, whereby the nucleotide at one or more positions in the source barcode is exchanged for one of the other 3 nucleotides.

In this context it is natural to use the Hamming distance as our distance metric since it counts the number of substitutions needed to convert one sequence to another. Formally, let l denote the sequence length and let S_i denote the sequence of read i. Furthermore, let $S_i[j]$ denote the nucleotide found at the jth position in S_i, where $j = 1, \ldots, l$. Then the Hamming distance between sequences S_a and S_b is given by,

$$h(S_a, S_b) = \sum_{j=1}^{l} I(S_a[j] \neq S_b[j]). \tag{2.1}$$

where I denotes the indicator function defined by,

$$I(S_a[j] \neq S_b[j]) = \begin{cases} 1 \text{ if } S_a[j] \neq S_b[j], \\ 0 \text{ if } S_a[j] = S_b[j]. \end{cases} \tag{2.2}$$

Since we are considering the general case when the barcodes are randomized DNA sequences, we can infer the structure of the data in the absence of errors. First we consider a pair of random DNA barcodes and analyse their theoretical proximity in sequence space. A DNA barcode consists of l random nucleotides and there are 4 nucleotides, A, C, T and G, that are equally likely to occur at each nucleotide position. It follows that if we have two barcodes, the probability that they have different nucleotides at a particular position is $3/4$. This is because there are $4^2 = 16$ combinations of nucleotide pairs and only 4 of these are matching pairs, corresponding to the cases AA, CC, GG and TT. Let H denote the discrete random variable counting the number of mismatches between two random barcodes (their Hamming distance). Assuming that the nucleotides are generated independently for each nucleotide position, it follows that,

$$H \sim Bin(l, 3/4). \tag{2.3}$$

The distribution of H is shown in Figure 2.1a for barcode length $l = 20$. We can see that two random barcodes are unlikely to be close in sequence space, with an expected Hamming distance of 15.

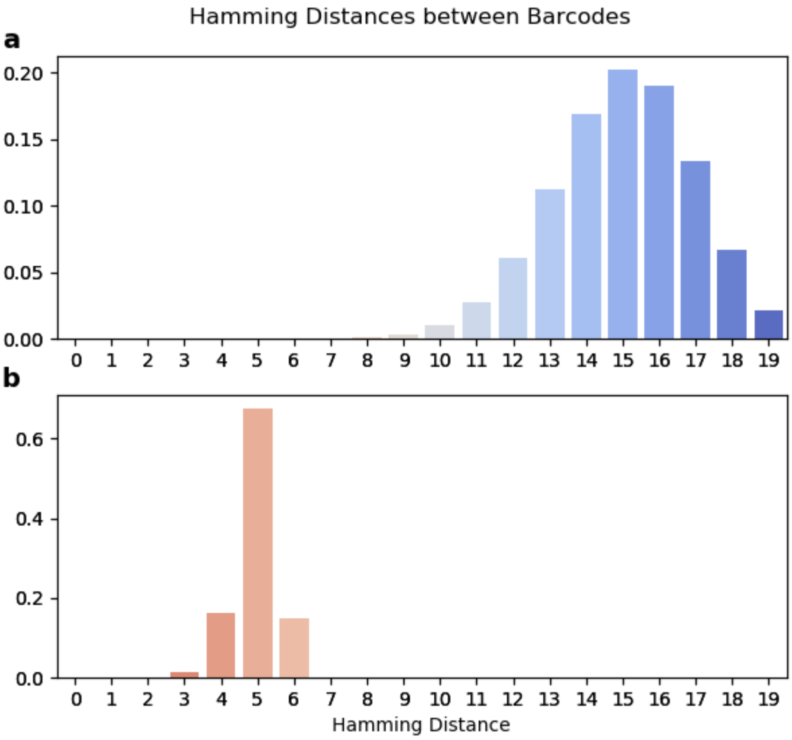


Figure 2.1: (a) The probability mass function of the random variable H for $l = 20$. (b) The probability mass function of Y for $m = 500\,000$ and $l = 20$.

It is tempting to conclude from this that all barcodes will be distant in sequence space and that we do not have to worry about confusing the error reads from one with the error reads from another. However, there is an important distinction to be made. In Figure 2.1a we considered the

Hamming distance between two arbitrary barcodes from the population. It is more informative to consider the distribution of the Hamming distance between a given barcode and its closest barcode in the population. If the sequence space is populated with a large number of barcodes, these distributions will differ greatly. The distribution of the hamming distance to the closest barcode is more relevant for our purpose, since we are often dealing with large numbers of barcodes and we want to know how likely it is for a given barcode to have another barcode in its close proximity.

Let m denote the number of barcodes in the population and consider a given barcode in the population. We want to know the distribution of the minimum Hamming distance to another barcode in the population. Let X_i denote the random variable counting the number of mismatches between the given barcode and the ith barcode. Note that X_i and H have the same distribution. This is because if one barcode is given the probability that it does not match another barcode at a specific nucleotide position is still $3/4$, just like the case when both barcodes are unknown.

The random variable $Y := \min(X_1, \ldots, X_{m-1})$ corresponds to the minimum Hamming distance between the given barcode and any other barcode. The random variables $X_1, \ldots, X_{m-1}$ are independent and identically distributed with cumulative distribution function,

$$F_X(x) = \sum_{k=0}^{x} \binom{l}{k} (3/4)^k (1/4)^{l-k}. \tag{2.4}$$

It follows that the CDF of Y is given by,

$$\begin{aligned} F_Y(y) &= P(Y \leq y) = P(\min(X_1, \ldots, X_{m-1}) \leq y) \\ &= 1 - P(X_1 > y, \ldots, X_{m-1} > y) \\ &= 1 - \prod_{i=1}^{m-1} P(X_i > y) \\ &= 1 - (1 - F_X(y))^{m-1}. \end{aligned} \tag{2.5}$$

Figure 2.1b shows the probability mass function of Y for $m = 500\,000$ and $l = 20$. We see that the most probable Hamming distance of a given barcode to its nearest barcode is 5. We also see that for some barcodes the closest barcode can have Hamming distance as small as 3 or 4.

To understand the data, we also need to consider the error reads and where they might fall in the sequence space in relation to their source barcodes. To do this we will use a simplified error model where we assume that errors occur with the same probability at each nucleotide position. While error probabilities can vary at different nucleotide positions in real sequencing data, this assumption simplifies the mathematical modelling of error reads greatly, and provides a reasonable approximation [9]. We will also assume that once an error has occurred in one position it does not become more or less likely for an error to occur at another position (independence). We will set the error probability per nucleotide to 0.24% in accordance with the experimentally determined estimate provided by Pfeiffer et al. in [8]. Let E denote the discrete random variable counting the number of errors that occur when a barcode is read. Under our error model the distribution of E is given by,

$$E \sim Bin(l, 0.0024). \tag{2.6}$$

Since the error probability is low in this case, around 95% of the reads are error free under this model. The probability that 1 error occurs in a barcode is 4.6% and the probability that 2 errors occur is 0.1%. To understand what effect these errors might have, we need to account for the total number of reads. We will assume that the average frequency of a barcode in the population is 100 and that the number of barcodes is 500 000 as before. Then the expected number of total reads is $100 \times 500\,000 = 5 \times 10^7$. Since the probability that one error occurs is 4.6×10^{-2}, it follows that the expected number of reads with 1 error is $4.6 \times 10^{-2} \times 5 \times 10^7 = 2.3 \times 10^6$. Using the same reasoning, the expected number of reads with 2 errors is $10^{-3} \times 5 \times 10^7 = 5 \times 10^4$. This is assuming that each read is an independent Bernoulli trial with success probability 4.6% and 0.1% respectively.

As we have already demonstrated the barcodes themselves can sometimes be close in sequence space, within 3 or 4 substitution in some cases. If 1 or 2 errors occur in a barcode so that the error read is brought closer to the closest neighboring barcode it can be difficult to determine which barcode it originated from. Some of these cases are almost impossible to resolve, even in theory. Especially when the closest barcode to an error read is not its source barcode. More commonly, the closest barcode is the source barcode but another barcode is also in close proximity to the error read. To resolve these cases the clustering procedure needs to identify all nearby barcodes and assign the sequence to the closest one. To understand why, consider a simple clustering procedure, based on the idea of merging an error read with the first identified barcode within some distance threshold to it. Since there might be some other barcode that is even closer than the first one encountered this strategy would lead to errors in such cases. The procedure we propose in section 3 is able to resolve these cases by considering all putative barcodes within a distance threshold before merging. In contrast, the most recent method proposed by Zhao et al. in [10] merges an error read with the first viable barcode within a distance threshold. A statistical test is used to determine the viability of the barcode as the source barcode of the error read based on the Hamming distance between the reads and their frequencies.

In practice there are additional challenges. First we have to correctly identify each sequence as either a true barcode or a potential error read. High frequency sequences (e.g. above frequency

20) are almost certainly true barcodes, since the probability that many error reads end up with the same sequence is negligibly small. When the frequency of the sequence is low, it is harder to determine and in those cases we need to consider its proximity to other sequences.

3 Method

In this section we will explain each step performed by our method to find accurate barcode clusters. In section 3.1 we introduce the k-mer index which forms the backbone of our algorithm and enables us to efficiently find a neighborhood for each sequence containing all nearby sequences. In section 3.2 we introduce the main clustering procedure that utilizes the k-mer index. Section 3.3 details the Bayesian hypothesis test that we use in ambiguous cases to determine if a sequence is a true barcode or an error read. Finally, the procedure used for automatic parameter selection is described in section 3.4, where we also discuss performance optimization.

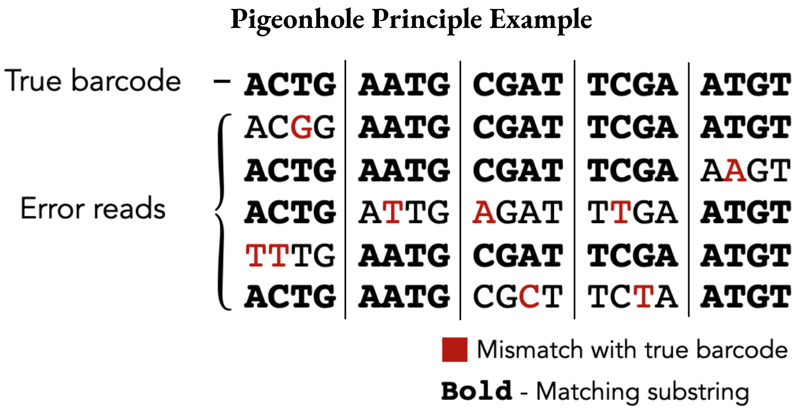
3.1 The k-mer index

The idea behind the k-mer index is that if we can efficiently find neighborhoods for the reads, without the need to compute the Hamming distances between all pairs of unique reads, we can start to make decisions about which reads to group together. Given a read we define its ϵ-neighborhood as the set of all reads within Hamming distance ϵ to it. In section 3.4 we will detail how ϵ is determined. For now our task is to efficiently find the ϵ-neighborhood of each read.

To do this we start by thinking of each read as a series of non-overlapping substrings of length k called k-mers, a well known concept in bioinformatics. As before let l denote the barcode length. Although it is not generally required, we will assume that k divides the read evenly to illustrate the idea more clearly. Since the barcode length is fixed each choice of k partitions the read into a number of k-mers, let $p = l/k$ denote the number of partitions.

Given a distance threshold ϵ, we require that k is chosen so that $p > \epsilon$. If a read is within distance ϵ of a given read we claim that the number of k-mers shared by the reads is at least $p - \epsilon$. This follows from the pigeonhole principle exemplified in Figure 3.1. From the figure we see that all error reads are within distance $\epsilon = 3$ of the true barcode. Consequently, each error read will share two or more 4-mers with the true barcode. This is because we have a maximum of three errors that can occur in five 4-mers. Placing one error in each 4-mer minimizes the number of matching 4-mers, but will always leave two 4-mers error free.

It follows from this principle that for a given read the set of reads that share with it $p - \epsilon$ or more k-mers is guaranteed to include all reads in its ϵ-neighborhood. We will call this set the k-mer



Pigeonhole Principle Example

True barcode	**ACTG**	**AATG**	**CGAT**	**TCGA**	**ATGT**
Error reads	ACGG	**AATG**	**CGAT**	**TCGA**	**ATGT**
	ACTG	**AATG**	**CGAT**	**TCGA**	AAGT
	ACTG	ATTG	AGAT	TTGA	**ATGT**
	TTTG	**AATG**	**CGAT**	**TCGA**	**ATGT**
	ACTG	**AATG**	CGCT	TCTA	**ATGT**

Figure 3.1: Illustration of the pigeonhole principle for $l = 20$, $k = 4$ and $\epsilon = 3$. Each error read is within Hamming distance 3 of the true barcode. Consequently, it follows from the pigeonhole principle that each error read will share 2 or more 4-mers with the true barcode.

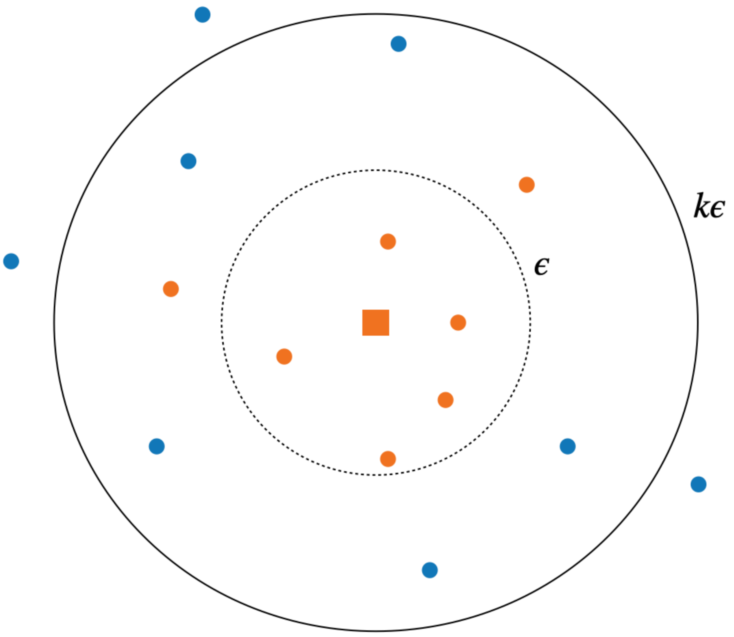
neighborhood of the read. Note that reads outside of the ϵ-neighborhood might also be included in the k-mer neighborhood. However, since $p - \epsilon$ k-mers match, the k-mer neighborhood will not include any reads with Hamming distance more than $l - k(p - \epsilon) = l - l + k\epsilon = k\epsilon$. An illustration of the read neighborhoods is shown in Figure 3.2.

There are $\binom{p}{p-\epsilon}$ ways for two reads to share $p - \epsilon$ k-mers. Equivalently, there are $\binom{p}{p-\epsilon}$ ways to choose $p - \epsilon$ k-mers of a read. The k-mer index is a lookup table that maps each one of these k-mer combinations, found in at least one of the unique reads in the data, to the set of all reads that share it. To construct the index we iterate over the unique reads and in each iteration we perform the following steps:

1. Find the $\binom{p}{p-\epsilon}$ combinations of k-mers for the read.

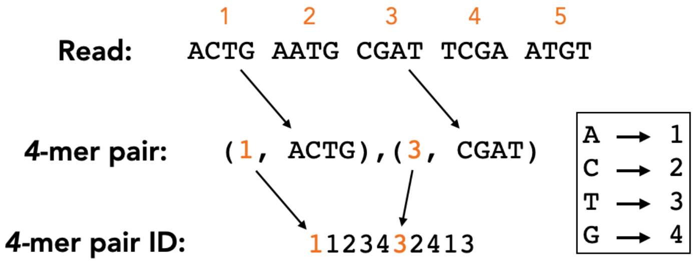
2. Convert each of these combinations to an identification number (ID) that will serve as the key for that combination. This process is illustrated in Figure 3.3.

3. Is the ID already in the k-mer index?
   No: Add the key and map it to a set containing the current sequence.
   Yes: Add the current sequence to the set of sequences sharing the key.



Read Neighborhood Illustration

Figure 3.2: A given read (orange square) surrounded by its neighbors (dots) in sequence space. The orange dots are the k-mer neighbors of the read, i.e. all sequences that share at least $p - \epsilon$ k-mers with it. The blue dots are reads not included in the k-mer neighborhood. The dotted circle is the ϵ-neighborhood of the read and the solid circle is the boundary for the k-mer neighbors, i.e. no k-mer neighbor will appear outside of the solid circle. Note that all ϵ-neighbors of the read are also k-mer neighbors, this is guaranteed by the pigeonhole principle.



k-mer Combination to ID Conversion

1 2 3 4 5
Read: ACTG AATG CGAT TCGA ATGT
4-mer pair: (1, ACTG),(3, CGAT)
4-mer pair ID: 1123432413
A → 1
C → 2
T → 3
G → 4

Figure 3.3: Illustration of how a pair of 4-mers are converted into a combination ID. First a pair of 4-mers is selected. Each 4-mer has a location in the read specified by the orange numbers. Then the 4-mer pair is transformed into an ID by mapping each of its nucleotides to a number specified by the conversion table in the bottom right of the figure.

Once these steps have been performed for each sequence the k-mer index has been constructed. Given a read we can now use the index to find its k-mer neighborhood. To do this we simply access its combination IDs obtained from step 2 of constructing the index. Then we query the index using these IDs to find all reads that share at least one ID, these reads are the k-mer neighbors of the read.

So far we have seen how the k-mer index is constructed and how it can be used to find k-mer neighborhoods. Once we have the k-mer neighborhood of a read we can find its ϵ-neighborhood by computing the Hamming distances between the read and all of its k-mer neighbors, only keeping the reads that are within distance ϵ gives us the ϵ-neighborhood. However, it is still unclear how using our method for finding ϵ-neighborhoods is more efficient than computing all pairwise Hamming distances between the reads to do so. The fundamental reason that our approach is significantly more efficient is that constructing the k-mer index only requires one pass over the reads. In contrast, using the naive approach of computing all pairwise distances requires iterating over the reads, and in each iteration another pass over all reads is required (excluding the given read). In essence, the work required to construct the k-mer index is roughly equivalent to computing the distances between just one read and all other reads. However, once constructed, the index enables us to efficiently find k-mer neighborhoods, narrowing the search for ϵ-neighbors considerably compared to searching all reads for ϵ-neighbors.

There are two parameters that need to be fixed before the k-mer index can be constructed, the distance threshold ϵ and the substring length k. In section 3.4 we will detail the procedure used for determining these parameters.

3.2 Clustering Procedure

In this section we will explain the simple clustering procedure used to identify true barcodes and to group them with the error reads that originated from them. The procedure is based on the observation that the high frequency reads are unambiguously true barcodes while low frequency reads may be error reads. We assume that the k-mer index has already been constructed for some distance threshold ϵ and substring length k. The procedure works as follows. We iterate through the sequences in descending order of frequency and use the k-mer index to find the k-mer neighborhood of each read. We start by classifying the first read as a true barcode since it is the highest frequency read. Subsequent reads are also classified as true barcodes as long as none of their ϵ-neighbors are true barcodes. The rationale behind this is that if the read was an error read we would expect its source barcode to have a higher frequency and to have been identified as a true barcode in a previous iteration. In that case we would expect it to appear in the ϵ-neighborhood of the sequence.

On the other hand, if a true barcode is found in the ϵ-neighborhood of the read under consideration the read could be an error read that originated from the true barcode. However, when the barcode length is short (e.g. $l = 20$) true barcodes can be close to each other in sequence space as demonstrated in section 2. Therefore, the naive strategy of classifying a read as an error read whenever it has a true barcode as an ϵ-neighbor is subject to type II errors.

To resolve these cases statistically, we will use a Bayesian hypothesis test to determine if a read is an error read or a true barcode. This test will be described in detail in the next section. For now we will treat it as a step if the algorithm which takes as input the read to be examined and its closest true barcode. It also accounts for the frequencies of both sequences and the estimated per nucleotide error rate, from the processes of PCR amplification and sequencing. The output is a classification of the read under consideration as either a true barcode or an error read originating from its closest true barcode. If the read is classified as an error read, it will be clustered with its closest true barcode. In the rare cases when two or more true barcodes have the same Hamming distance to the read under consideration, the one with the highest frequency will be considered. The clustering procedure is summarized below in Algorithm 1.

Algorithm 1 Sequence Clustering Procedure

```
Input: sequences (including their frequencies), $k$-mer index, $\epsilon$, error rate estimate $\rho$
Output: clustered sequences
true_barcode_set = $\emptyset$
sorted_sequences = sort_by_frequency(sequences, order=descending)
for seq in sorted_sequences do
    neighborhood = getNeighbors(seq, $k$-mer index)
    true_barcode_neighbors = true_barcode_set $\cap$ neighborhood
    if true_barcode_neighbors $\neq \emptyset$ then
        closest_true_barcode = getClosest(seq, true_barcode_neighbors)
        if h(seq, closest_true_barcode) $\leq \epsilon$ then
            class = hypothesis_test(seq, closest_true_barcode, $\rho$)
            if class == error_sequence then:
                cluster seq with closest_true_barcode
                continue to next iteration
    add seq to true_barcode_set
```

The clustering procedure we have described performs a statistical test every time a true barcode is in the ϵ-neighborhood of the sequence under consideration. We want to avoid performing this test many times for cases that are unambiguous to save computational time. In section 3.4 we will describe how the statistical test can be used before iterating through the sequences to find simple rules to identify such cases. It is important to note that these optimizations will only affect the computational time of the procedure and do not affect the clustering result.

3.3 Bayesian Hypothesis Test

When a given read is within Hamming distance ϵ of a true barcode, we have to decide if it should be classified as an error read or a true barcode. The decision should account for the sequences of both reads and their frequencies together with the estimated average per nucleotide error rate ρ. The parameter ρ is the estimated probability that a substitution error occurs at a nucleotide in a read. In section 3.4 we explain how this parameter is determined.

Let S_c denote the sequence being considered and let f_c denote its frequency. Furthermore, let S_b denote the sequence of the neighboring true barcode with frequency f_b. The Hamming distance between the sequences is given by d, such that $d \leq \epsilon$.

We will consider two competing models for the read. In the first model, M_1, the read S_c originated from the nearby barcode S_b through substitution errors. In the second model, M_2, the read

S_c is itself a true barcode generated independently of the nearby true barcode S_b. The marginal likelihood of each model M_i takes the form,

$$P(f_c, S_c \mid S_b, f_b, \rho, M_i). \tag{3.1}$$

Naturally this marginal likelihood will depend greatly on the model we are considering.

3.3.1 Marginal Likelihood of Model M_1

Let us start by considering model M_1. To find a computable expression for the marginal likelihood of this model we will use the probability chain rule to obtain,

$$P(f_c, S_c \mid S_b, f_b, \rho, M_1) = P(f_c \mid S_c, S_b, f_b, \rho, M_1)P(S_c \mid S_b, \rho, M_1), \tag{3.2}$$

where we have used $P(S_c \mid S_b, f_b, \rho, M_1) = P(S_c \mid S_b, \rho, M_1)$, i.e. the probability of observing the read S_c only depends on the sequence of the nearby barcode S_b, but not on its frequency f_b. We can think of $P(S_c \mid S_b, \rho, M_1)$ as the probability of converting the sequence S_b to S_c in one trial/reading. From this point of view it is clear that while f_b is directly related to the total number of trials, given by the true frequency of S_b in the population, it does not affect the probability that S_b is converted to S_c in one of these trials.

Given this interpretation we can also find a computatable expression for $P(S_c \mid S_b, \rho, M_1)$. Each time an error occurs at a nucleotide position, there are 3 nucleotides to replace the correct one. We will assume that each one of these 3 possibilities is equally likely. Since the distance between S_b and S_c is given by d it follows that the probability of converting S_b to S_c in one trial is estimated by,

$$P(S_c \mid S_b, \rho, M_1) = \hat{p}_{bc} = (\rho/3)^d(1-\rho)^{l-d}, \tag{3.3}$$

where we have assumed that the error rate is the same at each nucleotide position, and that errors occur independently at each nucleotide position. We will elaborate on the validity of our assumptions in section 5.

We can see that $\hat{p}_{bc}$ is normalized by summing over all possible sequences S_c to obtain,

$$\begin{aligned}\sum_{S_c} P(S_c \mid S_b, \rho, M_1) &= \sum_{k=0}^{l} \binom{l}{k} 3^k(\rho/3)^k(1-\rho)^{l-k} \\ &= \sum_{k=0}^{l} \binom{l}{k} \rho^k(1-\rho)^{l-k} = 1.\end{aligned} \tag{3.4}$$

The term 3^k appears in the first equality since there are 3 possible nucleotides at each of the k positions where the two sequences differ.

An expression for $P(f_c \mid S_c, S_b, f_b, \rho, M_1)$ is also needed so that we can compute the marginal likelihood of model M_1. It is the probability of observing the frequency f_c for the read S_c given that it originated from the neighboring true barcode S_b with the observed frequency f_b. We assume that S_b has some unobserved true frequency in the population that we will denote n. We can think about the process of sequencing S_b as n independent Bernoulli trials, each one has a probability $\hat{p}_{bc}$ (Eq. 3.3) of converting S_b to S_c. From this point of view f_c follows a binomial distribution with parameters n and $\hat{p}_{bc}$, and we want to evaluate the probability of observing f_c under this model. To proceed, the unobserved parameter n needs to be estimated.

Let us consider the distribution of f_b. Since we are assuming that S_b is a true barcode, it follows that f_b is its observed frequency in the population. In particular, it is the number of times S_b was read without errors. The probability of no error occurring in one reading of S_b is estimated by $\hat{p}_{ne} = (1-\rho)^l$. We can now think of f_b as a sample from a binomial distribution with parameters n and $\hat{p}_{ne}$. Consequently, we can obtain the maximum likelihood estimate of n given by,

$$\hat{n}_{mle} = \left\lfloor \frac{f_b}{\hat{p}_{ne}} \right\rfloor. \tag{3.5}$$

where $\lfloor \cdot \rfloor$ denotes the floor function. A proof that this is the maximum likelihood estimate is presented in Appendix A. For a more thorough discussion on the estimation of this parameter we refer to [2]. Under the current model, M_1, both reads originated from the same source barcode and so we need to ensure that our estimate of n is not less than $f_b + f_c$. Therefore, our estimate of n is given by,

$$\hat{n} = \max(\hat{n}_{mle}, f_b + f_c). \tag{3.6}$$

Using this estimate we obtain the following expression for the desired probability,

$$P(f_c \mid S_c, S_b, f_b, \rho, M_1) = p(f_c; \hat{n}, \hat{p}_{bc}) = \binom{\hat{n}}{f_c} \hat{p}_{bc}^{f_c} (1 - \hat{p}_{bc})^{\hat{n} - f_c},$$

where $p(k; n, p)$ denotes the probability mass function of a binomial distribution with parameters n and p evaluated at k. The marginal likelihood of our first model M_1 (Eq. 3.2) is now estimated by,

$$\begin{aligned} P(f_c, S_c \mid S_b, f_b, \rho, M_1) &= P(f_c \mid S_c, S_b, f_b, \rho, M_1) P(S_c \mid S_b, \rho, M_1) \\ &= p(f_c; \hat{n}, \hat{p}_{bc}) \hat{p}_{bc}. \end{aligned} \tag{3.7}$$

3.3.2 Marginal Likelihood of Model M_2

In a similar way we can also find an expression for the marginal likelihood of model M_2. Like before we use the chain rule to obtain,

$$P(f_c, S_c \mid S_b, f_b, \rho, M_2) = P(f_c \mid \rho, M_2) P(S_c \mid S_b, M_2). \tag{3.8}$$

In Eq. 3.8, we use the property that the frequency of the read under consideration, S_c, is independent of the nearby barcode S_b and its frequency f_b. However, since we know that S_c and S_b are distinct sequences, the probability of observing S_c will not be independent of S_b. Furthermore, the probability of observing S_c does not depend on the error rate ρ, since it is a randomized sequence under model M_2. On the other hand, the probability of observing f_c will depend on ρ. This becomes clear if we think about f_c as the number of times S_c was read without errors.

Since S_c is a random DNA sequence under M_2 with 4 possible nucleotides at each of the l positions it follows that,

$$P(S_c \mid S_b, M_2) = \frac{1}{4^l - 1} \approx \frac{1}{4^l}. \tag{3.9}$$

The probability $P(f_c \mid \rho, M_2)$ is more difficult to determine. It is the probability of observing the frequency f_c for a true barcode S_c in the population, given the error rate ρ. Since the observed frequency distribution of the reads includes error reads, the distribution of the observed true barcode frequencies is unknown. What we do know is the maximum observed frequency f_{max}. Given this maximum, we have a range for the possible frequency values between 1 and f_{max}. With no additional information we want to assume as little as possible about the frequency distribution. This is achieved by choosing the maximum entropy distribution, given by the discrete uniform distribution in our case. It follows that for $f_c \in [1, f_{max}]$,

$$P(f_c \mid \rho, M_2) = \frac{1}{f_{max}}. \tag{3.10}$$

The marginal likelihood of model M_2 is now given by,

$$P(f_c, S_c \mid S_b, f_b, \rho, M_2) = P(f_c \mid \rho, M_2) P(S_c \mid M_2) = \frac{1}{4^l f_{max}}. \tag{3.11}$$

3.3.3 Read Classification using Bayes Factor

We are now able to compute the marginal likelihood of each model. To compare the models and to determine which model describes a given read better we will use the log Bayes factor, $\ln K$, the logarithm of the ratio between the marginal likelihoods of M_1 and M_2 given by,

$$\begin{aligned} \ln K &= \ln P(f_c, S_c \mid S_b, f_b, \rho, M_1) - \ln P(f_c, S_c \mid S_b, f_b, \rho, M_2) \\ &= \ln p(f_c; \hat{n}, \hat{p}_{bc}) + \ln \hat{p}_{bc} + l \ln 4 + \ln f_{max}. \end{aligned} \tag{3.12}$$

To make a decision about whether the current read is an error read or a true barcode we will consider M_1 as our null model. We only want to reject this model if its marginal likelihood is significantly lower than the marginal likelihood of the alternative model M_2. By default, we will reject the null model if $\ln K \leq -4$, i.e if the marginal likelihood of model M_2 is approximately 55 times greater than the marginal likelihood of model M_1. This threshold was chosen in accordance with the guidelines provided in section 3.2 in [4]. The threshold can be adjusted by the user to control the trade-off between type I and type II errors. Increasing the threshold will increase the number of type I errors while reducing the number of type II errors.

3.4 Parameter Selection and Optimization

In this section we will discuss how we can determine the parameters of our procedure automatically based on the observed data. We will also discuss how our clustering procedure can be optimized to improve performance. Our algorithm relies primarily on three parameters, the maximum distance ϵ for merging reads, the substring length k used for k-mer indexing and the total error rate per nucleotide ρ.

3.4.1 Determining ϵ

We will start by fixing ϵ appropriately. We want to choose ϵ so that the vast majority of error reads are within the ϵ-neighborhoods of their source barcodes. However, we do not want to choose a larger ϵ than necessary. Firstly, the memory and time complexity of the algorithm increase for larger values of ϵ. This is because a larger ϵ necessitates that the sequence is divided into more partitions with smaller k, since we require that the number of partitions $p > \epsilon$. In general, this will increase the number of k-mer combinations and consequently the number of entries in the k-mer index. As a result, the index will take up more space in memory and will take longer to construct. Another reason that we want to avoid choosing a larger ϵ than necessary is that the k-mer neighborhoods become larger as ϵ increases. Since we search the k-mer neighborhoods for true barcodes this leads to an increase in search time. If the distance between S_c and S_b is large enough our statistical test will classify S_c as a true barcode regardless of the frequencies of the reads. Therefore a reasonable choice for ϵ is the largest distance such that S_c could still be classified as an error read for some frequency setting. To find this distance we start by considering the frequency setting that will clearly favour model M_1 for a given distance d between S_c and S_b. We set $f_b = f_{max}$ to favour model M_1. It remains to find the value of f_c that maximizes the marginal likelihood of model M_1. From Equation (3.7) we see that maximizing the marginal likelihood of model M_1, with respect to f_c, is equivalent to finding the mode of the binomial distribution with parameters $\hat{n}$ and $\hat{p}_{bc}$. However, since $\hat{n}$ is a function of f_c we will replace it with $\hat{n}_{mle}$ to determine the mode. We can do this since we know that the value of f_c that maximizes the marginal likelihood of model M_1 will be much smaller than $f_b = f_{max}$, since it represents the most likely frequency of an error read originating from sequence S_b. Consequently, it is safe to assume that $\hat{n}_{mle} > f_b + f_c$, which implies $\hat{n} = \hat{n}_{mle}$. Since $f_c > 0$ it follows that given $f_b = f_{max}$ and a distance d between S_c and S_b, the value of f_c that maximizes the marginal likelihood of model M_1 is given by,

$$f_c = \max\left(\lfloor(\hat{n}_{mle} + 1)\hat{p}_{bc}\rfloor, 1\right). \tag{3.13}$$

To find ϵ we apply our statistical test using the frequency combination f_c (as defined in 3.13) and $f_b = f_{max}$, for increasing values of d. The largest value of d for which S_c is classified as an error read will be chosen as the value for ϵ.

3.4.2 Determining k

When choosing k we need to make sure that $p > \epsilon$. However, in most cases there are several choices of k that satisfy this constraint. On the one hand, we want to choose a small k so that $k\epsilon - \epsilon$ is small, this corresponds to the distance between the dashed circle and the solid circle being small in Figure 3.2. Since we only consider ϵ-neighbors for merging, this ensures that the number of irrelevant reads in each neighborhood with distance greater than ϵ are minimized. This will decrease the size of each neighborhood resulting in shorter search times. However, as we mentioned previously a smaller k will also increase the number of k-mer combinations, increasing the memory use and running time of the algorithm.

To find a reasonable value for k we will only consider the true barcodes in the absence of errors. The reason for this is that error reads will be close to their source barcodes in sequence space. Consequently, if we focus on excluding true barcodes, that are not associated with a given true barcode, we are simultaneously excluding many of the error reads of those distinct barcodes as well. We will also assume that ϵ has already been fixed. It should be noted that the optimal value for k depends on the hardware used for running the algorithm. However, our approach here does not consider the hardware and only attempts to find a reasonable choice based on the theoretical distribution of Hamming distances for random barcodes.

For a given barcode we want to ensure that the number of distinct barcodes in the region between the dashed circle and the solid circle in Figure 3.2 is small. As discussed in section 2 the Hamming distance from a given barcode to a random barcode follows a binomial distribution with l trials and success probability $3/4$. Given this distribution we will require that,

$$P(\epsilon < d \leq k\epsilon) = \sum_{d=\epsilon+1}^{k\epsilon} \binom{l}{d} \left(\frac{3}{4}\right)^{d} \left(\frac{1}{4}\right)^{l-d} < \frac{1}{2}. \tag{3.14}$$

This constraint ensures that for a given barcode the majority of distinct barcodes are expected to be either within distance ϵ or beyond distance $k\epsilon$. This ensures that we do not pick a k that is too large. Given (3.14) and the constraint $p > \epsilon$ we will now pick the largest integer that satisfies both as our value for k. The constraint (3.14) is somewhat arbitrary since there is no inherent reason to choose $1/2$ as our threshold. Nevertheless, we have chosen it here since it resulted in appropriate choices for k in practice. For example, our parameter selection scheme found $k = 4$ to be the optimal value for both datasets considered in section 4. For both datasets the barcode length was $l = 20$ and $\epsilon = 3$ was chosen. By considering the other two possible values for k, 2 and 5, we can see why this choice is optimal. If $k = 2$, the k-mer index will take up too much space in memory. If $k = 5$, the index takes up slightly less space in memory compared to $k = 4$. However, this choice increases the k-mer neighborhood size significantly, since $k\epsilon = 5 \times 3 = 15$ for $k = 5$ compared to $k\epsilon = 4 \times 3 = 12$ for $k = 4$. This might seem like a small difference at first but if we consider the hamming distance distribution for true barcodes (see figure 2.1a) we see that many of the distinct barcodes will be within hamming distance 12 to 15.

It is important to note that the parameter k only affects the memory usage and running time of the algorithm and has no effect on the clustering results. In fact the only parameters that impact the clustering results are ϵ and ρ. Specifically, the ϵ parameter dictates which sequences are considered for merging and ρ is used in the statistical test to determine if a sequence is an error read or a true barcode.

3.4.3 Determining ρ

To determine ρ we need to consider the design of the barcode sequences. So far we have considered the case when each nucleotide in a barcode is random. In general, real barcode designs include constant regions that separate the random regions of the barcodes. These constant regions have nucleotides that are shared by all barcodes. An example of a typical barcode design is one with 20 random nucleotides separated by 3 constant regions with 2 consecutive nucleotides each. In this design the first 5 nucleotides are random followed by 2 constant nucleotides and then 5 random nucleotides again. This pattern repeats until the number of random nucleotides is 20. The final configuration of the barcode is described by the numeric string 5-2-5-2-5-2-5. This design was used in [5] and we will use it in the next section for our simulated datasets. Some barcode designs, such as the one used in [7], do not include constant regions between the random regions. However, regardless of design there will always be constant regions that flank the barcodes on either end. These constant regions are present so that the barcodes can be located for PCR amplification and sequencing.

For the purpose of clustering we are only interested in the random regions, since the constant regions do not provide information to distinguish two barcodes. However, we can use these constant regions to obtain an estimate of ρ. Since we know what nucleotides should be present in these regions, we can identify any error that occurs due to PCR amplification or sequencing. Consequently, we can find the fraction of errors in each of the constant nucleotide positions. While the error rate might differ at different nucleotide positions, we simply assume that the average per nucleotide error rate can provide a good approximation. To estimate ρ we average the fraction of errors at each constant nucleotide position. Averaging is performed since some nucleotide positions have higher error rates than others, particularly the ones closer to the end of the sequences. Due to phasing effects that we will discuss further in section 5. As a result of these effects, it is important to choose constant regions that are spread out across the sequences to obtain a good estimate of ρ.

3.4.4 Performance Optimization

The statistical test described in section 3.3 is important for classifying a sequence in cases when it is unclear whether it is a true barcode or an error read. However, a simple threshold for the distance d or the frequency f_c will suffice to classify the read accurately in many of the cases encountered. By introducing appropriate thresholds to identify these cases we can avoid performing the statistical test repeatedly, which leads to a reduction in computational time. There are primarily two common classes of reads that we want to focus on for performance optimization.

The first common case is that the read has a high enough frequency that regardless of how close it is to a neighboring barcode, it will still be very likely to be a true barcode. For these cases we want to find a high frequency threshold f, such that any read with frequency $f_c \geq f$ is more likely to be a true barcode than an error read. To do this we consider the case when we have a read S_c, with the smallest nonzero Hamming distance $d = 1$ to the true barcode S_b. Furthermore, we consider the case when $f_b = f_{max}$. The idea is to maximize the marginal likelihood of model M_1. To find f we perform our statistical test for increasing values of f_c, starting from the value of f_c given in equation (3.13). We are looking for the smallest value of f_c for which the marginal likelihood of model M_2 is greater than the marginal likelihood of M_1. Consequently, the first value of f_c such that the statistical test classifies the read S_c as a true barcode will be chosen as the frequency threshold f. Once we have obtained f we can classify all reads with frequency $f_c \geq f$ as true barcodes, without having to perform the test again for each of these cases.

There is also another case that we want to deal with separately to decrease the running time of our algorithm. Many of the error reads will have frequency 1. This is because most errors that originate from the same barcode are unique under reasonable assumptions on the error rate, the barcode length and the barcode frequency distribution. To save time we want to find a distance threshold, τ, such that any sequence with frequency $f_c = 1$ that is within Hamming distance τ to a barcode S_b is more likely to be an error read than a true barcode, regardless of the frequency of S_b. As f_b increases, the likelihood that the current read is a true barcode decreases. Because of this we will now consider the case when $f_b = 1$, which is when S_c has the highest likelihood of being a true barcode for a given distance d from S_b. We will now start with distance $d = 1$ and perform our statistical test for increasing values of d. The largest value of d for which S_c is classified as an error read will be chosen as the value for τ. Any read with frequency 1 within distance τ of its barcode neighbor can now be classified as an error read.

Finally, we can save some computational time when computing the Hamming distances between a read and its nearby true barcodes. Since we are not interested in barcodes beyond distance ϵ we will use the truncated Hamming distance. For sequences S_a and S_b with Hamming distance d, their truncated Hamming distance is given by,

$$h_t(S_a, S_b) = \begin{cases} d \text{ if } d \leq \epsilon, \\ l \text{ otherwise.} \end{cases} \tag{3.15}$$

Using the truncated Hamming distance allows us to save time by stopping the computation of the Hamming distance once it has exceeded ϵ.

4 Results

In this section we will evaluate the accuracy and performance of our approach on simulated datasets. We will compare it to the state-of-the-art method for barcode read clustering, Bartender, which was detailed in [10]. To simulate the datasets we will follow the procedure used in [10] with some modifications to the simulation parameters.

4.1 Simulated Datasets

The methods will be compared on two simulated datasets. Both of them will have 500 000 true barcodes, each with 20 random nucleotides and 3 constant spacers with 2 nucleotides each in the configuration 5-2-5-2-5-2-5. These choices were made to imitate the real dataset produced in [5].

Once the 500 000 barcode sequences have been generated, each one is assigned a frequency by drawing a sample from an exponential distribution with mean 100 and set the frequency of the barcode to be the least integer greater than or equal to the sample (the ceiling function). To simulate errors we consider the process of reading each barcode as many times as it appears in the population. Furthermore, we will assume a constant per nucleotide error rate.

Each time a barcode is read, we perform a Bernoulli trial at each of its nucleotide positions with the chosen error rate as the probability of success. When one of these trials is successful, an error has occurred and the nucleotide at that position is replaced with one of the other 3 nucleotides with equal probability. Once the errors have been introduced, the simulated datasets consist of a set of unique reads and their frequencies. If a barcode was destroyed in the error generating process, i.e., if every time it was read an error was introduced, all reads associated with that barcode were removed from the dataset. This was done to simplify the evaluation since destroyed barcodes that have been clustered correctly are difficult to distinguish from false positives.

The chosen error rate for dataset A is 0.33% per nucleotide and the error rate for dataset B is 0.66%. The error rate for dataset A was chosen to be close to the estimated error rate found in [8]. This estimation was based on the Illumina sequencing platform [1]. A sequencing platform is a protocol for performing DNA sequencing and the Illumina platform is one of the most widely used protocols. Since different sequencing platforms have different error rates the error rate for dataset B was set to be twice the error rate of dataset A to account for this variation. The two simulated datasets are summarized in Table 4.1.

Dataset	A	B
Unique Read Count	3 433 217	5 773 822
True Barcode Count	499 640	499 320
Total Barcode length	26	26
Random Barcode length (l)	20	20
Error rate	0.33%	0.66%

Table 4.1: Summary of each simulated dataset

4.2 Evaluation

Both methods were tested using their default settings when possible. This was done to mimic the results a user would obtain without rerunning the algorithms and adjusting the parameters. For our method the parameters were automatically determined as described in section 3.4 with the default threshold for log Bayes factor (-4). For Bartender, the maximum Hamming distance considered for merging two sequences is a user defined parameter with no default. We set this parameter to match the ϵ parameter determined by our method on the same dataset. The motivation for this is that both parameters control the maximum Hamming distance considered for merging. By setting the same value for both methods, we ensure that none of the methods are at a disadvantage from being more restricted in the merging process.

For both datasets our method automatically set $\epsilon = 3$ and $k = 4$. The parameter ρ was estimated from the constant regions of the barcodes and matched the chosen error rate for each dataset with less than 0.02% error in both cases. The auxiliary parameter τ was set to 2 for both datasets. Finally, the frequency threshold f was set to 17 for dataset A and 22 for dataset B.

We will focus on three accuracy measures to compare the clustering results of each method: the false positives, the false negatives and the percentage error of the estimated barcode frequencies. We will define a false positive as any detected cluster that does not contain a true barcode. Similarly, we define a false negative as a true barcode that was clustered together with a higher frequency true barcode. To compare the percentage error in the estimated frequencies, we will only consider the true barcodes correctly identified by both methods.

A comparison of the number of false positives and false negatives for each method on each dataset is shown in Figure 4.1. We can see that our method has substantially lower counts for both measures on dataset A. In particular, the false positive counts for Bartender is almost 2 orders of magnitude higher when compared to our method on this dataset. Higher false positive counts result in more distortion of the relative barcode frequencies. Specifically, it leads to a large number of spurious low frequency lineages, since most false positives are small groups of error reads from a common source barcode. We see that the results are similar for dataset B but the difference in the false positive counts between the methods are less dramatic.

False negatives correspond to lineages that were missed and false positives are spurious lineages. As a consequence of these errors, the estimated frequencies of the barcodes that have been correctly identified will also have errors in some cases. To evaluate these errors, we consider all true positives identified by both methods, i.e., all clusters that contain a true barcode found by both methods. If more than one true barcode belongs to the same cluster, the one with the highest frequency represents the cluster. The frequency sum of a cluster estimates the frequency of the true barcode that represents it.

Accuracy Metrics Count Comparison

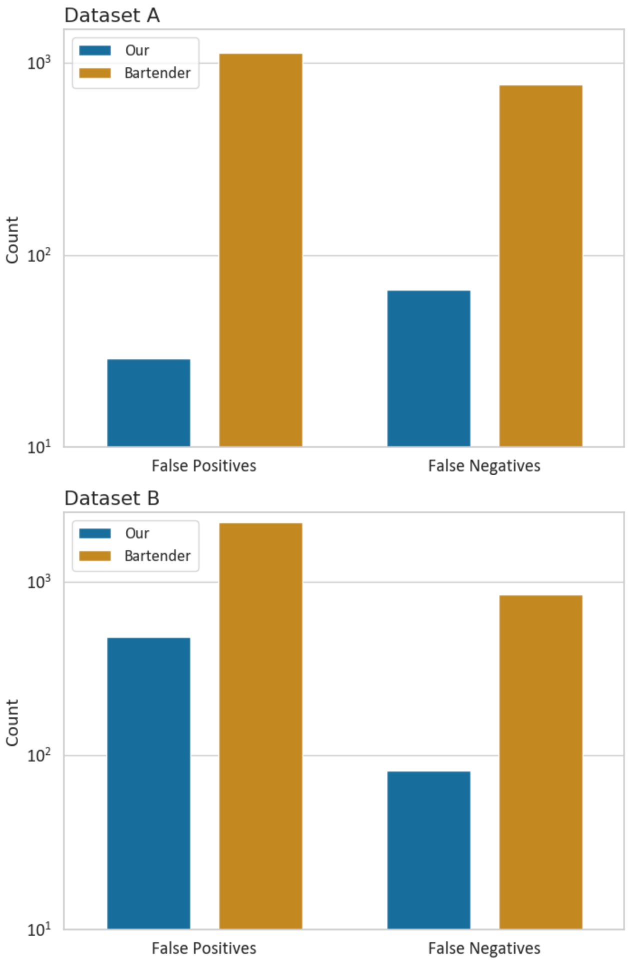


Figure 4.1: Count comparison of false positives and false negatives for each method on datasets A and B.

To determine the extent to which each method distorts the true barcode frequencies, we evaluate the empirical cumulative distribution function (eCDF) of the percentage error in their frequency estimates. Formally, let N denote the number of true positives shared by both methods. Furthermore, let Z_i denote the random variable corresponding to the percentage error in barcode i. Assuming that Z_1, Z_2, ..., Z_N are independent and identically distributed (i.i.d.) the eCDF is given by,

$$\hat{F}_N(z) = \frac{1}{N}\sum_{i=1}^{N} I(Z_i \leq z). \tag{4.1}$$

The i.i.d. assumption does not strictly hold in our case. For example, error reads from one barcode can be incorrectly merged with another barcode and so the percentage errors are not independent. However, since we are not trying to make inference about the true CDF or quantities that depend on it, we will allow this contravention. Figure 4.2 shows the eCDF for each method on datasets A and B. For dataset A the number of shared true positives is $N = 498\,867$ and for dataset B we have $N = 498\,474$. For both datasets we see that Bartender is considerably more error prone than our method. In particular, we are concerned about high percentage errors that cause substantial distortion of the true frequencies. We see that Bartender has at least 3% error for 1 in 200 lineages on dataset A and for 1 in 100 lineages on dataset B. Since we have roughly 500 000 shared true positives in each dataset this corresponds to approximately 2500 barcodes in dataset A and 5000 barcodes in dataset B. In contrast, we see that the number of barcodes with a higher percentage error than 1% is negligibly small for our method on both datasets.

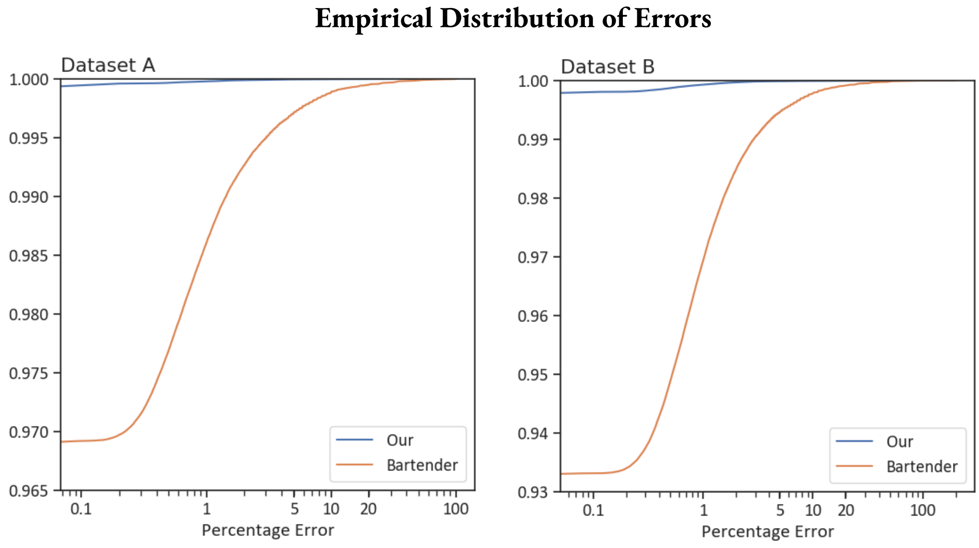


Figure 4.2: Empirical cumulative distribution of percentage errors in frequency for each method on datasets A and B.

5 Discussion

In summary, the method presented here for the task of barcode read clustering exploits the pigeonhole principle to efficiently find neighborhoods for each sequence. By estimating the per nucleotide error rate for PCR amplification and sequencing, we developed a Bayesian hypothesis test that accurately classifies each sequence as either a true barcode or an error read. The method was shown to be significantly more accurate on simulated datasets than Bartender, the most recent method proposed for the task of barcode read clustering.

To interpret the results shown in the last section and to understand how our method makes a significant contribution to the field, we need to consider how barcodes are used in practice. Note that we are only considering a single time point here. In practice, barcodes are used to track the relative frequencies of lineages over time. This is done by performing PCR amplification and sequencing at different time points. In [10] Bartender was used to cluster each time point independently and a merging procedure was used to connect clusters at subsequent time points. Because each time point is clustered independently in Bartender the single time point results shown here can shed light on the multiple time point accuracy as well.

Specifically, the frequency errors shown in Figure 4.2 for Bartender are introduced at each time point. Meaning that a series of errors could occur at different time points in the same lineage. This might cause the frequencies of the lineages to increase or decrease if errors in subsequent time points act in the same direction. This variance in the frequencies of some lineages induced by error could present difficulties for understanding the evolutionary dynamics of a population. In particular, it could make it difficult to distinguish between frequency fluctuations due to errors and frequency changes due to natural selection or genetic drift. Future work will focus on extending the single time point procedure presented here to multiple time points, where the higher accuracy of our approach can offer higher resolution lineage tracking than what is currently possible.

To enable the estimation of error rates using constant regions or flanking regions, a few simplifying assumptions were made that we will address here. Firstly, we estimate the per nucleotide error rate with the scalar ρ. The implicit assumption behind this is that the average error rate is the same at each nucleotide position in a barcode. In general, this assumption does not hold since phasing effects are a common issue with next-generation sequencing. These effects occur when some sequences among the PCR copies of each barcode to fall out of sync with the other copies during the PCR amplification process. There will be more sequences out of sync once the sequencer has reached the end of the sequence and therefore we often observe higher error rates at the end. For a more thorough review of phasing effects and error rates in next-generation sequencing, we refer the reader to [8]. Since we use constant regions and flanking regions for error rate estimation, we can only estimate the error rate directly at those selected constant nucleotide positions. Consequently, it is not possible to use the constant regions to directly estimate the error rate at each nucleotide position of the barcode.

Many sequencing platforms include quality scores for each read that estimate the sequencing error probability separately at each nucleotide position [3]. It is important to note that these quality scores do not account for the error rate associated with PCR amplification. We considered the possibility of using these quality scores to obtain separate error rate estimates for each nucleotide position. However, this idea was abandoned since it would limit the applicability of the method to those sequencing platforms that provide quality scores. Nevertheless, using quality scores as a way of estimating the error rate at each nucleotide position is a promising direction for future work that seeks to increase the resolution of lineage tracking further.

Another assumption made is that the errors at different nucleotide positions are independent i.e., the occurrence of an error in one nucleotide position does not affect the probability of an error in another. To the best of our knowledge, there is no evidence that such interaction effects exist.

To make the idea behind our method more clear, we assumed that the substring length k divides the barcode length l evenly. This assumption can be relaxed by allowing the division of a barcode into substrings of different lengths. A natural generalization is obtained by only requiring that $l = kq + r$ instead of $l = kq$ as before. For a given k that does not divide l evenly, the first q substrings can have length k and the last substring can have length r. Note that r is the remainder obtained from dividing l by k and so $r < k$. In this scheme, the largest Hamming distance that excludes a sequence from the neighborhood of another is no longer $k\epsilon$. It is now given by $k(\epsilon - 1) + r$. This corresponds to the case when one error occurs in the length r substring and each of the other $\epsilon - 1$ errors occur in different substrings of length k. The reason that we want to consider this generalization is that in some cases a choice of k that does not divide l evenly provides a better trade-off between k-mer index size and the number of irrelevant reads included in each k-mer neighborhood. Consequently, we are sacrificing performance if we limit ourselves to k for which k divides l evenly in such cases.

Because our method considers every putative barcode in the neighborhood of a read before merging, it can be naturally extended to incorporate the concept of fuzzy clustering. In fuzzy clustering every data point is assigned a membership probability to each identified cluster. This is in contrast to hard clustering, where each data point is assigned unanimously to a single cluster. As we showed in section 2 the ϵ-neighborhoods of nearby barcodes can overlap. Meaning that error reads from one of the barcodes can be close to the other barcode in sequence space. If a read is equally close to two different barcodes a fuzzy clustering would highlight our uncertainty about its membership. We could use this information to obtain confidence intervals for the barcode frequencies as oppose to just point estimates.

Finally, in this work we have only evaluated the accuracy of our approach on simulated datasets. In future work we aim to extend the evaluation to datasets that have been generated from real barcode experiments.

A Appendix

Here we will prove that the maximum likelihood estimate in equation (3.5) is correct by following a similar approach to the ones outlined in [2].

Theorem. *Let X be a random variable with a binomial distribution for which the trial success probability $p \in (0, 1]$ is known and the number of trials n is unknown. Given a single sample x from this distribution the maximum likelihood estimate of n is given by,*

$$\hat{n}_{mle} = \left\lfloor \frac{x}{p} \right\rfloor . \tag{A.1}$$

Proof. The likelihood function is given by,

$$\mathcal{L}(n; p, x) = \binom{n}{x} p^x (1-p)^{n-x}.$$

We will consider the ratio of the likelihood functions for successive trial counts given by,

$$r(n) = \frac{\mathcal{L}(n+1; p, x)}{\mathcal{L}(n; p, x)} = \frac{n+1-p(n+1)}{n+1-x}.$$

This ratio is non-increasing as a function of n since,

$$r'(n) = \frac{(1-p)(n+1-x) - (n+1-p(n+1))}{(n+1-x)^2} = \frac{x(p-1)}{(n+1-x)^2} \leq 0.$$

The inequality follows since $x(p-1) \leq 0$ for all $p \in (0, 1]$ and the denominator is positive for all valid values of n and x satisfying $n \geq x$. Since $r(n)$ is non-increasing the smallest integer n for which $r(n) < 1$ maximizes the likelihood function. Note that $r(n) < 1$ is equivalent to,

$$p(n+1) > x \iff n+1 > \frac{x}{p}.$$

It follows that the maximum likelihood estimate of n is the smallest integer satisfying $\frac{x}{p} < n+1$ which proves that the estimate in (A.1) is indeed the maximum likelihood estimate. □

Bibliography

1. D. R. Bentley et al. "Accurate whole human genome sequencing using reversible terminator chemistry". *Nature* 456:7218, 2008, pp. 53–59. ISSN: 1476-4687. DOI: 10.1038/nature07517. URL: https://doi.org/10.1038/nature07517.

2. S. Blumenthal and R. C. Dahiya. "Estimating the Binomial Parameter n". *Journal of the American Statistical Association* 76:376, 1981, pp. 903–909. ISSN: 01621459. URL: http://www.jstor.org/stable/2287586.

3. B. Ewing and P. Green. "Base-calling of automated sequencer traces using phred. II. Error probabilities". *Genome research* 8:3, 1998, pp. 186–194. ISSN: 1088-9051. DOI: 10.1101/gr.8.3.186. URL: https://doi.org/10.1101/gr.8.3.175.

4. R. E. Kass and A. E. Raftery. "Bayes Factors". *Journal of the American Statistical Association* 90:430, 1995, pp. 773–795. ISSN: 01621459. URL: http://www.jstor.org/stable/2291091.

5. S. F. Levy, J. R. Blundell, S. Venkataram, D. A. Petrov, D. S. Fisher, and G. Sherlock. "Quantitative evolutionary dynamics using high-resolution lineage tracking". *Nature* 519:7542, 2015, pp. 181–186. ISSN: 1476-4687. DOI: 10.1038/nature14279. URL: https://doi.org/10.1038/nature14279.

6. L. V. Nguyen, D. Pellacani, S. Lefort, N. Kannan, T. Osako, M. Makarem, C. L. Cox, W. Kennedy, P. Beer, A. Carles, M. Moksa, M. Bilenky, S. Balani, S. Babovic, I. Sun, M. Rosin, S. Aparicio, M. Hirst, and C. J. Eaves. "Barcoding reveals complex clonal dynamics of de novo transformed human mammary cells". *Nature* 528:7581, 2015, pp. 267–271. ISSN: 1476-4687. DOI: 10.1038/nature15742. URL: https://doi.org/10.1038/nature15742.

7. A. N. Nguyen Ba, I. Cvijović, J. I. Rojas Echenique, K. R. Lawrence, A. Rego-Costa, X. Liu, S. F. Levy, and M. M. Desai. "High-resolution lineage tracking reveals travelling wave of adaptation in laboratory yeast". *Nature* 575:7783, 2019, pp. 494–499. ISSN: 1476-4687. DOI: 10.1038/s41586-019-1749-3. URL: https://doi.org/10.1038/s41586-019-1749-3.

8. F. Pfeiffer, C. Gröber, M. Blank, K. Händler, M. Beyer, J. L. Schultze, and G. Mayer. "Systematic evaluation of error rates and causes in short samples in next-generation sequencing". *Scientific Reports* 8:1, 2018, p. 10950. ISSN: 2045-2322. DOI: 10.1038/s41598-018-29325-6. URL: https://doi.org/10.1038/s41598-018-29325-6.

9. M. Schirmer, R. D'Amore, U. Z. Ijaz, N. Hall, and C. Quince. "Illumina error profiles: resolving fine-scale variation in metagenomic sequencing data". *BMC Bioinformatics* 17:1, 2016, p. 125. DOI: 10.1186/s12859-016-0976-y. URL: https://doi.org/10.1186/s12859-016-0976-y.

10. L. Zhao, Z. Liu, S. F. Levy, and S. Wu. "Bartender: a fast and accurate clustering algorithm to count barcode reads". *Bioinformatics* 34:5, 2017, pp. 739–747. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btx655. eprint: https://academic.oup.com/bioinformatics/article-pdf/34/5/739/25458376/btx655.pdf. URL: https://doi.org/10.1093/bioinformatics/btx655.

11. E. Zorita, P. Cuscó, and G. J. Filion. "Starcode: sequence clustering based on all-pairs search". *Bioinformatics* 31:12, 2015, pp. 1913–1919. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btv053. eprint: https://academic.oup.com/bioinformatics/article-pdf/31/12/1913/566334/btv053.pdf. URL: https://doi.org/10.1093/bioinformatics/btv053.