Methods for Finding Structural Variations in DNA Barcodes

Johannes Isaksson

Department of Astronomy and Theoretical Physics
Lund University

May 2016

Bachelor thesis supervised by Tobias Ambjörnsson and co-supervised by Saair Quaderi
Abstract

Rapid spread of antibiotic resistance is currently one of the greatest threats to human health according to the World Health Organization [1] [2]. It is directly connected to horizontal gene transfer of plasmids, wherein resistance-coding genes often appear as structural variations (SVs). Therefore, developing comprehensive and rapid tools for detecting SVs is of great importance in both clinical biology and genomic research.

Currently, many of the established DNA sequencing techniques require hours or even days [3] [4] to run and are consequently, in some instances, unfit for clinical settings. This thesis presents and compares three approaches for rapid identification of SVs based on optical DNA mapping, two of which are new to this thesis. Although not yet adequate for application in their current state, we justify the usage of some of the techniques in the novel algorithms. We also demonstrate how one of the techniques shows some promise and is open for future amends.

Populärvetenskaplig sammanfattning


Ett viktigt skäl till varför antibiotisk resistens kan sprida sig så snabbt är gener som ligger i bakteriers väldigt mobila och dynamiska delar av arvsmassan; så kallade plasmider. Plasmider är små cirkulära strängar av DNA som effektivt kan kopieras och överföras bakterier sinsemellan. De kan dessutom förhållandevis enkelt inkorporera eller skapa nya gener; en process som i detta sammanhang kallas för en ”strukturell variation”. Skulle en


Att kunna analysera DNA-sekvenser på olika sätt är grundläggande för djupare förståelse för naturen och inte bara applicerbart på bakteriell resistens mot antibiotika. Utan metoder för att analysera DNA hade vi exempelvis inte kunnat veta att vi har omkring 98% DNA gemensamt med schimpanser och ca 50% gemensamt med bananer. Detta kanske i och för sig inte verkar som de mest nödvändiga bitar av information men de utgör likväl exempel på vetenskapens framsteg inom området. Sjukdomar som Alzheimers, cancer, och Parkinsons m.m. har alla sin egen koppling till DNA och att förstå dessa kopplingar kan vara avgörande för att hitta potentiella botemedel. Steget från att veta att vi har 50% DNA gemensamt med bananer till att finna lösningen till den postantibiotiska eran kanske trots allt inte är så stort som vi tror.
Contents

1 Background ........................................................................................................................................ 4
  1.1 Bacterial DNA & Plasmids ........................................................................................................... 4
  1.2 From DNA Sequences to DNA Barcodes ..................................................................................... 4
  1.3 Structural Variations .................................................................................................................... 5
2 Roadmap & Overview ...................................................................................................................... 6
3 Dynamic Time Warping (DTW) ......................................................................................................... 7
  3.1 Classic DTW .................................................................................................................................. 7
  3.2 Variations of DTW ........................................................................................................................ 10
4 Implications of DTW Algorithms in DNA Sequence Comparisons .............................................. 11
  4.1 Problems with the Standard Variations of DTW .......................................................................... 12
5 The MVM Algorithm ....................................................................................................................... 13
6 Adapted LCMA Segment Detection ................................................................................................. 15
  6.1 Tuning the Parameters .................................................................................................................. 18
  6.2 Finding a One-to-one Alignment .................................................................................................. 16
7 Comparing the Methods .................................................................................................................. 21
8 Results ............................................................................................................................................... 22
  8.1 Dependence on Noise Level ......................................................................................................... 25
9 Conclusion and Outlook .................................................................................................................... 26
Appendix A: The Linear Assignment Problem Approach .................................................................... 28
Appendix B: The Phase Randomization Technique ............................................................................ 29
10 References ....................................................................................................................................... 30
1 Background

1.1 Bacterial DNA & Plasmids

Plasmids are small, circular DNA molecules which are part of a bacterium’s genome and are physically separate from the chromosomal DNA. Plasmids are typically between one and one thousand kilobase pairs (kbp) long and generally carry genes that are beneficial for the bacterium’s survival [5]. Some are known to include genes encoding for antibiotic resistance. In contrast, the chromosomal DNA is much larger and carries more essential information for living and reproducing. As opposed to the long chromosomal DNA, the plasmids are highly dynamic with respect to their DNA sequence. Moreover, they can be transferred in between different bacteria, even between different species, and can even replicate themselves [5]. Due to their highly dynamic nature, plasmids have been identified as one of the main reasons for the rapid spread of antibiotic resistance in bacteria [6].

1.2 From DNA Sequences to DNA Barcodes

In bioinformatics, so called DNA barcodes have become increasingly common as a tool to identify and distinguish between sequences of DNA [7] [8] [9]. It can loosely be viewed as a type of fingerprint for DNA molecules. Although there is a plethora of methods for sampling and analyzing DNA sequences, in this thesis the focus will be on an Optical DNA mapping technique developed by the Westerlund group at Chalmers University [10] [11] [12] [13]. Optical DNA mapping methods rely on fluorescence microscopy to visualize certain features of DNA molecules. The main benefit over other methods is the large read length of 100-1000 kbp, compared to about 200-1000 bp for typical DNA sequencing. However, it comes with a drawback of having a lower resolution (about 1 kbp). In principle, even longer DNA molecules can be examined, but the length is limited by current DNA extraction methods. Another important benefit is the quick speed at which sequences can be visualized. The particular method developed by the Westerlund group uses a competitive binding technique where they submerge the DNA strands in a solution containing fluorescent YOYO-1 molecules and non-fluorescent Netropsin molecules. Netropsin has a strong preference to bind to AT-rich regions whereas YOYO-1 has no preference at all. The result is a sequence-dependent fluorescence intensity along the DNA molecule where GC-rich regions are prone to higher intensity levels. This varying intensity along the length of the molecule is interpreted as a DNA barcode. A detailed schematic of the DNA barcode creation process is presented in Figure 1.
Figure 1: Schematic of the DNA barcode creation process from the Westerund group article, ref. [13]. 1. Plasmids are extracted from bacteria and stained from a solution of YOYO-1 and Netropsin, then inserted in a nanofluidic chip. 2. The sample is driven through the nano channels with pressure and the plasmids are identified (right panel). The circular plasmids are cut using light irradiation, then stretched to a linear state (left panel) 3. 200 images are taken of each plasmid and combined to create what is called a kymograph (pictured). 4. The kymograph is converted into an intensity curve and the procedure is repeated for several plasmids of the same type to create a so called "consensus" barcode.

Most of the barcodes that will be treated in this thesis are, however, computer generated from the exact known sequences of the plasmids of interest [10]. These so called “theory barcodes” are typically used to interpret experimental results, but are extensively used here due to a shortage of accessible experimental data. They are also an excellent tool for validation purposes since one can make changes in the underlying sequences and know exactly where the changes are. Theory barcodes are generated using a statistical framework in which the binding probabilities of the YOYO-1 and Netropsin molecules are known. Other parameters used to generate theory barcodes include the solution’s YOYO-1 and Netropsin concentrations, the camera resolution, and of course the nucleotide sequences. All of these are known by our experimental collaborators and the theory matches well with consensus barcodes, making it a reliable instrument.

1.3 Structural Variations

Structural variations (SVs) are variations in the genetic structure of an organism. When an organism acquires new genes, such as those coding for antibiotic resistance, they often arise in the form of structural variations in the organism’s genome. As such, finding structural variations within DNA sequences is of paramount importance in medicine and genetics. These variations are roughly classified into five major categories:

- Insertions/inserts: a gene has been inserted somewhere in the DNA
• Deletions: part of the DNA has been removed
• Inversions: part of the DNA has been inverted with respect to the rest
• Repeats: part of the DNA is repeated twice or more, often in succession
• Translocation: part of the DNA has been moved to another location

The size of these SVs vary widely, from about one kbp to several Mbp. Smaller random variations such as SNPs (Single-Nucleotide Polymorphism) [14] do occur but are not recognized as SVs in this context. The large range of possible lengths constitutes intractably large computational problems if one attempts to find all of them. Due to the scale of the problem and the limited time, this thesis will focus exclusively on finding inserts. The methods presented here can, however, potentially be extended to deal with any type of structural variation. For instance, it is clear that when comparing two DNA sequences X and Y, an insert in X may be viewed as a deletion in Y, and vice versa, as long as whichever is the original is irrelevant.

The greatest problem when trying to find SVs in DNA from experimental data is the presence of noise and statistical variations which always occur in experimental settings. A further challenge when comparing experiment-vs-experiment or experiment-vs-theory barcodes is the uncertainty associated with the intensity values due to the stochasticity in the staining and visualization process. When kymographs are converted to intensity curves according to step 4 in Figure 1, the intensities are therefore rescaled as \( I = \frac{I - \langle I \rangle}{\sigma} \), where \( \langle I \rangle \) is the mean intensity and \( \sigma \) is the standard deviation, in order to have zero mean and unit variance. Another cause of such uncertainties are variations in intensity caused by temperature differences, but I refer to [12] for further details.

2 Roadmap & Overview

In the next section (section 3), background information will be given about an important algorithm in the field of comparative genomics, namely an alignment-algorithm called Dynamic Time Warping (DTW). Section 4 will go through how the algorithm is applied in the field and some of its limitations.

In section 5, we introduce the first new algorithm to the DNA barcoding field, called Minimum Variance Matching (MVM), which will be tested as a means of finding the previously mentioned structural variations. The algorithm was proposed by my supervisor and implemented from scratch by me. Section 6 introduces the second new algorithm, which we call Adapted LCMA (Length Constrained Minimum Average) Segment Detection, which will be used for the same purpose. The usage of the so called LCMA segments
was suggested by my supervisor but the adapted algorithm was developed by me. Section 6 also discuss some modifications and predicaments regarding the algorithm’s parameters. Existing software that will be used for the work in this thesis are theory barcode creation methods and a special method for generating random barcodes. The coding environment is Matlab.

Section 7 serves as a preface to the result section where some important remarks are made that contextualize the results such that one can compare the methods in a meaningful way. In the result section, section 8, the performance of the MVM and Adapted LCMA segment detection methods will be compared to the DTW algorithm. The thesis ends with section 10 with some concluding remarks and an outlook over possible challenges for the future.

3 Dynamic Time Warping (DTW)

Dynamic time warping is a well-known dynamic programming technique aimed to find the optimal alignment between two (real-valued) sequences. Originally developed as a speech recognition algorithm [15] for time-dependent queries, it has also been widely used in comparative genomics [16] [17] pattern discovery [18], and several other data mining applications. Intuitively, the DTW process corresponds to a form of global alignment procedure where sequences are locally warped in a nonlinear fashion to match each other in the best way.

3.1 Classic DTW

For two real-valued sequences $X = (x_1, \ldots, x_N)$ and $Y = (y_1, \ldots, y_M)$, in this case corresponding to intensity values of two barcodes with length $N$ respectively $M$, the DTW algorithm finds a warping path, $p$, going over all elements in $X$ and all elements in $Y$, such that a cost function $c(X, Y)$ is minimized. Naturally, $c(X, Y)$ is small if $X$ and $Y$ are similar to each other, and otherwise large. The task is then to find an optimal warping path, $p^*$, having the minimal cost, which denotes the optimal alignment between $X$ and $Y$. The process of finding $p^*$ is described more concretely below.

The first step is to evaluate the local cost measure for each pair of elements in sequences $X$ and $Y$. This is done by compiling the $N \times M$ Cost Matrix (henceforth referred to only as CM) where each index is archetypally the Euclidean distance from $x_n$ to $y_m$, i.e. $CM_{nm} = \sqrt{x_n - y_m^2}$. The result is a cost-landscape where the optimal alignment between $X$ and $Y$ runs along valleys from $CM_{11}$ to $CM_{NM}$. Like [15] [18] [19], we will also
use the Euclidean distance as the cost measure. In the context of DNA barcode comparisons, the Euclidean distance is simply the absolute difference in rescaled intensity.

From the CM, an Accumulated Cost Matrix (ACM) is put together according to

\[
ACM_{nm} = CM_{nm} + \min \left\{ ACM_{n-1,m-1}, ACM_{n-1,m}, ACM_{n,m-1} \right\}
\]

(1)

As such, the ACM cells contains the minimum cost to reach that cell, when traveling across the CM according to the restrictions above, and adding up the steps costs. A warping path, \( p = (p_1, \ldots, p_L) \), is then a consecutive sequence of ACM coordinates, \( p_i = (n_i, m_i) \), with \( p_{i+1} \in \{(n_i + 1, m_i + 1), (n_i + 1, m_i), (n_i, m_i + 1) \} \). Note that this definition implies a step size condition with the consequence that a path can only take diagonal, down or right steps with size one. Additionally the condition to go over all elements in \( X \) and all elements in \( Y \) in the classic DTW algorithm implies the requirement \( p_1 = (1,1) \) and \( p_L = (N, M) \). The total cost of a warping path is simply \( c(p) = \sum_{t=1}^{L} CM(p_t) \). Furthermore, the optimal warping path, \( p^* \), is the path that minimizes \( c(p) \), which connotes \( c(p^*) = ACM(N,M) \).

Examples of the CM and ACM when comparing two noisy theory barcodes of the R100 plasmid are shown in Figure 2, together with the optimal warping path \( p^* \) in red. This figure, however, uses a variation of DTW where the \( p_1 = (1,1) \) and \( p_L = (N, M) \) boundary conditions are not forced (see section 3.2). The barcodes themselves are also shown. As expected, the optimal warping path for two very similar sequences is very close to a straight diagonal line. Keep in mind that, even if two sequences are identical, there will still be a great deal of low cost regions outside of the diagonal warp path\(^1\). These regions will be referred to as “spurious matches” and constitute a major problem in DNA sequence and DNA barcode analysis.

\(^1\) Mathematically, this is not strictly true, but it holds for all sequences of interest other than the very trivial monotonous sequences. In principle, the number of spurious matches increases with the number of features of the sequence.
Figure 2: Illustration of the Subsequence Dynamic Time Warping (sDTW) process. (A) Cost matrix (CM) and optimal warping path $p^*$ (red) resulting from running the DTW algorithm to compare the sequences X and Y. Here X is the theoretical barcode for the R100 plasmid and Y is a noisy version of X cyclically permuted and repeated twice. Dark regions correspond to high similarity and bright regions corresponds to low similarity. (B) Accumulated Cost Matrix (ACM) from the cost matrix in (A) with optimal warping path $p^*$ in red. Each cell holds the minimum cost for a path to reach that particular cell. Dark regions corresponds to a low cost and bright regions to a high cost. Note that since the barcodes being compared are very similar, the optimal warping path is close to a perfect diagonal line. (C) The barcode sequences plotted before and after alignment. Each pixel corresponds to 541 base pairs.
3.2 Variations of DTW

Many possible modifications can be made to the classic DTW algorithm for a variety of reasons, such as decreasing computational time or acquiring better constrained warp paths. Such modifications include global path constraints, local step weights, and step size conditions, among others [20]. Due to limitations in time and scope, only a few of the many potential adjustments will be explored, focusing on those most relevant to this project.

Subsequence DTW (sDTW):

In most applications, one is not interested in the exact global alignment, but rather in whether one of the entire sequences is similar to a subsequence of the other sequence. In such a case sDTW can be used to find the optimal local alignment and hence the subsequence of Y which best corresponds to X. In practice this is done by allowing the path to start and end at any given position of the first and last row in the ACM respectively. This is actually what was done in the DTW process in Figure 2. The values of the last row in the ACM is called the distance function, as it denotes the minimum DTW distance to reach the given positions in Y. Since the ACM hold the minimum cost to reach each cell, the optimal path $p^*$ is computed recursively, starting from the minima of the distance function.

The concept of sDTW can be extended to find the $n$ best fitting subsequences by disallowing the next path to start too close to the previous paths. This can be implemented by forcing each path to start at different minima in the distance function. If two minima happen to be too close, one can set the immediate neighborhood of the first minimum in the distance function to infinity.

Local weights:

It may be of interest to favor either direction when computing the optimal path. In that case one can simply introduce local weights, $\omega_d$, $\omega_r$, $\omega_l$, for the different directions i.e. replace equation (1) by:

$$ACM_{nm} = CM_{nm} + \min \left\{ \omega_d ACM_{n-1,m-1}, \omega_l ACM_{n-1,m}, \omega_r ACM_{n,m-1} \right\} \quad (2)$$

Obviously, an equally weighted case reduces to classical DTW. However, it is important to note that even in that case, there is a preference for the diagonal direction since one diagonal step is effectively equivalent to one horizontal and one vertical step. For this
reason, weights of values (2,1,1) are often chosen [20]. In this project, the weights are set to (1,1,1) because of its simplicity and the fact that a diagonal alignment is preferred when searching for structural variations (if the barcodes are exactly the same, the path will be a perfect diagonal line).

Global constraints & Bands:
Global constraints can be used to limit the temporal skew between the sequences such that, for instance, one value of sequence X do not get matched to all values of sequence Y and vice versa. To visualize this, picture an ‘L’-shaped path running along the first column and the last row in the ACM. This is of course undesirable in the DNA matching methodology since an alignment like that holds no practical significance. To counteract this, one implements regions in the ACM from which the desired path cannot deviate. A common method for achieving this is to implement a “Sakoe-Chiba band” [15], which limits a path originating in \((x_n, y_m)\) to follow \(|(x_n - x_k) - (y_m - y_k)| \leq R\) for all step indices \(k\) and a specific width constraint \(R \in \mathbb{R}\). The result is a diagonal band with a width of \(2R + 1\), in which all paths must reside.

4 Implications of DTW Algorithms in DNA Sequence Comparisons

Because of the circular nature of the plasmids (see sect. 2.1) that are investigated in this project, special care needs to be taken into account when performing alignment between plasmid DNA barcodes. When the plasmids are extracted and analyzed, they are irradiation-cut (see Fig. 1) at random positions as a result of random photo-nicking, making the barcodes cyclically permuted with respect to one another. An easy way to bypass this problem is to concatenate one of the sequences with itself and then performing the alignment. As an example, consider the sequences ‘A-C-G’ and ‘C-G-A’. It is clear that the sequences are cyclically permuted versions of each other, but the DTW algorithm does not automatically account for this. On the other hand, when comparing the sequences ‘A-C-G-A-C-G’ and ‘C-G-A’, the algorithm realizes the similarity since C-G-A is now a subsequence of the other (‘A-C-G-A-C-G’).

Since the aim in this project is finding structural variations, it is important to consider how these would appear in a DTW-centered approach. To help with this, Figure 3 shows a sketch of how different SVs would look in the CM or ACM. It is apparent that, in order to detect a repeat, the algorithm requires mapping of one value of \(X\) to several values of
Y. Detection of inserts/deletions, on the other hand, only requires horizontal/vertical jumps.

![Diagram of structural variations](image)

**Figure 3:** Schematic illustration of how different structural variations (SVs) would appear as low cost regions in the CM (cf. Figure 2) when comparing sequence X with Y. The filled lines represents the ideal matches (i.e. optimal path) between the sequences and the dashed lines are disjointed parts resulting from the SVs.

### 4.1 Problems with the Standard Variations of DTW

While the DTW algorithms are very computationally efficient at finding optimal alignments, some problems may arise when blindly applying them to DNA barcodes, specifically when the presence of structural variations is of interest. If, for instance, there is an inserted gene in sequence Y (positioned horizontally) when comparing it to X (positioned vertically), one would want the path (going from top to bottom row of the ACM) to initially follow a diagonal route where the barcodes are the same, then follow a horizontal path where the insert is, and then follow a diagonal path where they are the same again (cf. Figure 3). However, there is no guarantee that the algorithm will find this ideal path and especially not for long inserts, since the path will have more opportunities to “derail” the longer the insert is. Furthermore, if the intensities of the inserted gene are very dissimilar to neighboring values of X, the right path will be associated with a very high cost when going over the horizontal gap. As such, modifications needs to be made for the algorithms to be better suited for finding these structural variations.

This thesis presents attempts to implement two different approaches to tackle these problems. The first one is to implement a partial shape matching technique developed by
Latecki et. al. [18] called Minimum Variance Matching (MVM). The second one is a method developed by me with inspiration from Park’s and Glass’s speech clustering paper [19], where a key element is the usage of so called Length Constrained Minimum Average (LCMA) segments [21]. This method will be called Adapted LCMA Segment Detection. Appendix A also show how the Linear Assignment Problem can be applied to address this problem, but that its effectiveness rapidly decreases with the presence of noise.

5 The MVM Algorithm

The MVM algorithm [18] was originally developed for the problem of partial shape matching, i.e. describing a shape and calculating its similarity to others. The main idea is to first convert the shapes into sequences of numbers then map the problem into finding an optimal path in a directed graph, which fulfills the same purpose as the ACM in the DTW algorithm. It can readily be applied to the problem of DNA barcodes since the barcodes are already represented in real valued sequences of intensity values. When comparing it to the DTW algorithm, it essentially implements an additional option where it allows the path to make horizontal jumps to the left, if the jumps are beneficial. Furthermore, as opposed to original DTW, MVM always makes a 1:1 mapping of X to Y values. This also implies that the path need not be continuous. The algorithm, which takes as input the length of X, N, length of Y, M, and the CM, is described in detail in Figure 4. Examples of paths found with the MVM algorithm for two theoretical barcodes is shown in Figure 11 in the result section. The key property that justifies the use of MVM is the ability to make horizontal jumps, bypassing inappropriate costs for inserts and thus improving the likelihood of finding the optimal path near an actual insert.

One expected issue with the MVM approach is that if the algorithm makes an incorrect jump, it should be unable to jump back, or have problems awaiting the real path to catch up again. This can be problematic because, as pictured in Figure 2 (A), there may be several spuriously low-cost areas outside of the desired path, which will attract jumps to them.
Algorithm: Minimum Variance Matchmaking (MVM)

**Input:** N, M and the cost matrix CM

**Output:** Matrix pathcost containing the cost of the shortest path between each pair of cells, matrix path with which we can backtrack the shortest path leading to each cell

1: \[ \text{elasticity} = \min(N-M, \text{winWidth}) \]
2: for \( i=1:M \)
3: \hspace{1em} for \( j=1:N \)
4: \hspace{2em} pathcost(i, j) = \text{Inf} \]
5: \hspace{2em} path(i, j) = 0 \]
6: \hspace{1em} end \]
7: \hspace{1em} end \]
8: for \( j=1:\text{elasticity}+1 \)
9: \hspace{2em} pathcost(1, j) = \sqrt{CM_{ij}} \]
10: \hspace{1em} end \]
11: for \( i = 2 : M \)
12: \hspace{2em} stopk = \min(i - 1 + \text{elasticity}, N); \]
13: \hspace{3em} for \( k = i - 1 : \text{stopk} \)
14: \hspace{4em} stopj = \min(k + 1 + \text{elasticity}, N); \]
15: \hspace{4em} for \( j = k + 1 : \text{stopj} \)
16: \hspace{5em} if pathcost(i, j) > pathcost(i-1, k) + \sqrt{CM_{ij}} \]
17: \hspace{5em} \hspace{1em} pathcost(i, j) = pathcost(i-1, k) + \sqrt{CM_{ij}} \]
18: \hspace{5em} path(i, j) = k \]
19: \hspace{4em} end \]
20: \hspace{3em} end \]
21: \hspace{2em} end \]
22: end \]

*Figure 4:* The Minimum Variance Matchmaking (MVM) algorithm from ref. [18] for finding an optimal alignment path between two real-valued sequences by utilizing subsequence matching. N and M is the lengths of the two sequences being compared. winWidth is an optional parameter which limits the elasticity of the path if necessary. On line 13, there is a typo in ref. [18] that prevents the algorithm from running if the two sequences are of equal length. Here, it has been corrected by subtracting 1 to the minimum value of k.

The above algorithm uses the elasticity parameter as a constraint for the maximum jump length. In this project we would like the alignment path to be able to end at all Y-indices from 1 to N in order to cover all possibilities in a 1:1 mapping. The input value M should therefore be the length of sequence Y after concatenation with itself, in order to satisfy this for lines 8-9. Likewise the winWidth parameter was set to infinity. The algorithm could potentially be made to perform slightly better if the elasticity could be the difference in length before concatenation, but this is outside the span of this project.
6 Adapted LCMA Segment Detection

The idea with the adapted LCMA segment detection method is to divide the ACM into a certain number of diagonal bands, and then to find an individual optimal path within each band. Within these paths, the algorithm then searches for LCMA segments, i.e. segments of at least a specific length that have a mean less than a specific threshold value. This is straightforward since each step in a path has a certain cost associated with it. Lastly, the algorithm attempts to construct a final global alignment path with the information of the LCMA segments that hopefully bypasses “abnormalities” such as structural variations.

The bands constrain the maximum temporal skew allowed for the paths such that these do not stray too far from the ideal diagonal path. This is desirable since an overly large temporal skew should not exist for generally similar sequences differing primarily by a few SVs. The bands are implemented by respecting the Sakoe-Chiba constraint (see section 4.2) when calculating the ACM as well as when calculating the path. This prevents the cells in the ACM from accidentally being calculated from invalid routes crossing over the desired bands and resulting in artificially low estimates for certain path costs (recall that each cell in the ACM contains the minimum path cost to reach that cell, wherever the path is originating from). The Sakoe-Chiba constraint also needs to be respected when calculating the paths to ensure that they do not cross over to other bands.

Using bands in conjunction with LCMA segments also potentially allows for finding smaller sections within each path that have particularly low costs (i.e. are more similar). This is what we expect to see if a path is partially good because the plasmids are matching, but is then cut off due to the presence of a SV (cf. Figure 3). Also note that there are various ways to find these portions of alignment that are particularly similar to each other. We proceed with the LCMA method because it has been proven successful in research related to pattern discovery in speech [19], a subject akin to comparative genomics. The process of finding LCMA segments can be more closely described as follows:

For a sequence of real numbers \( S = (s_1, ..., s_N) \) and a length constraint parameter \( L \), the LCMA segment is a consecutive subsequence \( s = (s_i, ..., s_j) \) of \( S \), with length at least \( L \), such that the average of all \( s_n \) is minimized. In other words, we wish to find \( i' \) and \( j' \) that satisfy \( i', j' = \arg\min_{1 \leq i \leq j \leq N} \frac{1}{j - i + 1} \sum_{k=i}^{j} s_k, \forall j - i + 1 \geq L. \)

We extend this concept to include all subsequences of \( S \) whose mean costs fall below a value of \( T \). The impetus for this is the observation that a single LCMA segment for same-type plasmids is generally only a small portion of the plasmids that happen to align better
than the rest. In our problem, we would much rather find all segments that align “good enough” since we expect the full alignment for same-type plasmids to fit well. The CM, along with the paths and the LCMA segments, of two noisy R100 barcodes are shown in Figure 5.

![Cost Matrix](image)

**Figure 5:** Example of a CM with warping paths as thin red lines and LCMA segments in bold for two noisy barcodes of the R100 plasmid. The horizontal barcode is repeated twice and circularly permuted for illustration purposes. The bold LCMA segments indicate parts of the paths which fit particularly well. The best match is mapping X from Y-index 75 to 245, which can be seen as a fully bold diagonal line. This run used a band width of 5 pixels, length parameter $L=3$ and threshold parameter $T=0.3$.

### 6.1 Finding a Global Alignment

The assignment of defining an unambiguous alignment path from the information of the LCMA segments is a rather demanding one and there are, of course, several ways to address it. The rationale for doing this is to more easily compare the method to MVM, but it should be noted that the method presented here is only an exploratory approach and further research needs to be done on the subject in order to make any claims about the effectiveness and generalizability. An additional benefit of defining a global alignment path is that it allows for establishment of a probabilistic method, which is something we are interested in pursuing in the future. Since the method will be compared to MVM, it is convenient to use 1:1 mapping in this path as well. In fact, compared to a 1:many-and-many:1 mapping, the 1:1 mapping only saves the best match over the many indices, but loses detailed information about the stretching. This is acceptable since some control over
the stretch is already inherent within the bandwidth parameter. Furthermore, when interpreting the results, it is important to note that methods for finding a global alignment path are heavily dependent on the parameters used.

The method that was developed uses the fact that one would expect authentically matching segments of DNA to be longer than spuriously matching segments. However with the presence of noise, the noise can easily alter the expected LCMA segments, for instance by cutting them in half. The noise typically also increases the number of spurious LCMA matches and as a result the longest LCMA segments do not necessarily belong to the correct path. To help alleviate this issue, the total number of LCMA pixels per path was used rather than the longest segments. Statistically, this makes sense because one would expect the paths to have about as many spurious matches as one another. If, on the other hand, one path is partially correct, one expects additional LCMA pixels in the part that is correct. With this in mind the following (somewhat naïve) pseudo-algorithm was devised for the final alignment:

**Algorithm: Adapted LCMA Path**

**Input:** Warping paths and LCMA segments from the LCMA segment detection method, number of bands $n$ within which the optimal path is allowed to travel, length $N$ of sequence $X$

**Output:** An optimal alignment path $p^*$

1. add the $n$ paths with the most LCMA pixels to current list
2. for $i = 1; N$
3. choose the best matching pixel $p$ of pixels at row $i$ among
4. paths in list
5. add $p$ to $p^*$
6. end
7. end
8. end

**Figure 6:** Pseudo code for the process of defining a singular 1:1 path for aligning sequence $X$ to sequence $Y$ in the adapted LCMA segment detection method. An example of a cost matrix with warping paths and LCMA segments, which are used as input for the algorithm, can be seen in Figure 5.

The above algorithm freely allows jumps between the $n$ paths. In reality, this is not very reasonable. To begin with, a single pixel (about 540 bp in this case) is smaller than the shortest length of a typical gene. Furthermore, the algorithm often makes a lot of jumps, especially when noise is present, which is not apparent in real data. Therefore, some kind of jump penalty or smoothing function could be of use. Such additions was not implemented within the range of this project due to time limitations. Note that, in an optimal setting, with $n$ set to max, this algorithm would be able to find any type of SV
(save for inversions and repeats). This algorithm is also accompanied by a predicament regarding the choice of \( n \); if \( n \) is too high, the path will jump around to an unreasonable extent, and if \( n \) is too low there is an increased chance the algorithm will not find the correct bands. Throughout the rest of this thesis, a value of \( n = 4 \) was chosen because there will only be one SV, and since a value of 4 bypasses the case where the second- and third-best band has a high LCMA pixel count by coincidence.

6.2 Tuning the Parameters

The major problem with the LCMA segment detection method is the introduction of several new parameters, most notably the length and threshold of the LCMA search. The width of the bands is another parameter. In this project, the width of the bands was chosen to be equal to the longest barcode autocorrelation length. This ensures that the ACM values in different bands are sufficiently uncorrelated, as the autocorrelation is a measure of a signal’s correlation with itself as a function of distance over the signal (called time lags \( k \)). The autocorrelation \( r_k \), at lag \( k \), for a real sequence \( y \) of length \( \tau \), with a variance of \( r_0 \), is defined according to [22] as

\[
    r_k = \frac{1}{r_0(\tau - 1)} \sum_{t=1}^{\tau-k} (y_t - y)(y_{t+k} - y)
\]

and the autocorrelation length we define as the integral (or sum in this case) of the autocorrelation function as in ref. [23]. This quantity is actually not strictly positive, but in the case of a zero or negative autocorrelation length, one may choose to integrate only up to the first zero-crossing or until the autocorrelation falls below for instance two standard deviations. All barcodes in this thesis have positive autocorrelation lengths by the first definition.

The sample autocorrelation function of a theoretical barcode of the R100 plasmid with an artificial insert (see below) is shown in Figure 6. The autocorrelation length was calculated to 5 pixels and the bandwidth parameter was accordingly set to the same value.
**Figure 7:** Autocorrelation function for the R100 plasmid theory barcode with an artificial insert computed from eq. 3. The sum of the function is used as width parameter in the LCMA segment detection method and was in this case calculated to 5 pixels. The blue horizontal lines indicate boundaries for plus/minus two standard deviations.

To determine values of the length and threshold parameters $L$ and $T$ a parameter plot was done with two theoretical non-noisy barcodes of the same plasmid, of which one had a fabricated insert. The insert was made by taking the first roughly 20000 bp from the known sequence of the T100 plasmid and inserting them into one of the R100 plasmid sequences before generating the theoretical barcodes. This is much like how a real insert would appear in an experimental setting, with the obvious exception that one would not know its location or size beforehand. The size of roughly 20 kbp was chosen because our experimental collaborators have seen other SVs of similar size [13], although they have yet to find an insert. With this, the following score measure was devised since the exact pixel locations of the insert are now known:

$$Score = \frac{\text{total number of hits}}{\text{total number of LCMA pixels}} - \frac{\text{total number of misses}}{\text{length of sequence } X}$$  \hspace{1cm} (4)

An LCMA pixel is defined as a hit if the pixel’s location belongs to the set of positions in the correct path. The total number of misses is defined as the sum of pixels in the correct path that does not belong to any LCMA segment. This measure guarantees a high score if many of the LCMA pixels are in line with the correct path (first term). It also makes sure they hit as much of the correct path as possible, so to not only place a few pixels along the correct path (second term). The score plot for the length and threshold parameters is shown in Figure 7. Clearly a lower value of the length parameter $L$ seems to be dominant when it comes to this measure of performance. A lower value of $L$ is also
beneficial in general since an overly large \( L \) limits the size of the SVs the algorithm can detect (for instance, an \( L \) larger than the very first part of the path in Figure 3 would not be able to acknowledge that part as an LCMA-segment). From these results, the parameter values were chosen to \( T = 0.3 \) and \( L = 3 \) pixels and will be used for the remainder of this thesis.

![Figure 8: Plot of the score measure (eq. 4) as a function of the length (L) and threshold (T) parameters in the LCMA method. The score was evaluated for two non-noisy theory barcodes of the R100 plasmid of which one had an artificial insert. The parameters which gave the best score, \( L=3 \) and \( T=0.3 \) (arrow), were chosen throughout the rest of the thesis.](image)

From Figure 7 it is evident that coinciding low values for both \( L \) and \( T \) produce very bad scores. This is because a low \( T \) produces fewer LCMA pixels and a low \( L \) produces more spurious matches (cf. Figure 2 A: many of the spurious matches are either horizontal or vertical segments and will hence not match for large \( L \)). In general, higher \( T \)-values tend to worsen the score, which is due to the method accepting a lot of LCMA pixels that do not belong to the correct path, hence reducing the first term in eq. (4). It is still slightly worse than expected, however, since we expect both terms in eq. (4) to approach zero when \( T \to \infty \).

There is actually another parameter that is the Sakoe-width for the paths within each band. This width is different from the previously mentioned Sakoe-width in the way that this restricts the path width, as opposed to the band width. Without it, the maximum path width is restricted only by the band width. By limiting this parameter one can make sure paths within adjacent bands are not extremely similar (for instance to prevent two paths from hugging the same “wall” between their bands). However, we will not use this
parameter (i.e. set it to zero) because there is no clear need for having a restriction like this and since it may invalidate the score measure mentioned above by limiting the available pixels the paths can travel on.

7 Comparing the Methods

Previously, some researchers have used the cross correlation for comparison between sequences [13] [24], but the cross correlation is in fact embedded in the path cost since we use Euclidean distance, which when squared becomes \( c(n, m)^2 = \sum_{n,m} (X_n Y_m)^2 = \sum_{n,m} (X_n^2 + Y_m^2 + 2X_n Y_m), \) where the last term is the twice the un-normalized cross correlation with zero mean and unit variance. This is true for all barcodes studied in this thesis since they all have undergone zero mean and unit variance normalization (see section 2.3). A problem with a similarity measure like this is that if one of the methods have a lot of falsely labelled matches it may have a lower cost than the real path, since the reason a method makes a false match in the first place is because that match had a lower cost than the corresponding index of the real path.

In order to better mimic an experimental setting when comparing the methods, noise is applied to the barcodes before comparing them. However this nullifies the right-or-wrong label for the pixels of the correct path. Because of this, we modify the label and say that a pixel is a hit if the pixel is placed within a range of one autocorrelation length (±4 pixels) from the original correct path. This way the method is “forgiven” for doing slight misplacements but obvious mismatches are still retained as misses. Part of the result section will be dedicated to study how the fraction of hits is related to the applied noise level. Since more individual barcodes in a consensus barcode can be seen as reducing the noise, data like that could potentially help experimenters realize how many individual barcodes are needed for the consensus to reach a desired fraction of hits. Only measuring the fraction of hits is in reality not a very fair measure of performance when it comes to detecting SVs since a method can get a high fraction of hits for the majority of the path, then completely miss all the pixels after the SV. Because of this, a true or false tag is introduced which denotes whether the methods found the artificial insert (see section 6.2) or not. The case is said to be true if the algorithm places at least 70% of the hits before the insert as well as 50% of the hits after the insert and false otherwise. These values were chosen because they fit our subjective view of the boundary where the algorithms start to make clear misjudgments.

When applying noise to the barcodes we make use of so called phase-randomized barcodes. The phase randomization technique was introduced to the DNA barcoding field by
the Ambjörnson group in order to create realistically looking semi-random barcodes [13], and is described in detail in Appendix B. To create a noisy barcode \( P_{\text{new}} \), a phase-randomized barcode \( P_{\text{random}} \) of the same length is first created. A fraction of the phase-randomized barcode is then added to the original barcode \( P \) to create the noisy barcode according to:

\[
P_{\text{new}} = (1 - \alpha)P + \alpha P_{\text{random}}
\]

The parameter \( \alpha \) was tuned to be as large as possible while still maintaining a cross-correlation of about 0.9 between the original and the noisy barcode. The cross correlation target value of >0.9 is what we have experimentally found to be the case for same-type plasmids. The tuning was done by averaging the cross correlation for 100 generated noisy barcodes for each value of \( \alpha \) and then plotting the average cross correlation as a function of \( \alpha \) and thereafter determining the 0.9 boundary. The result from this process is shown in Figure 9 below, which justifies a value of \( \alpha = 0.138 \).

![Cross correlation vs Alpha parameter](image)

**Figure 9:** Average cross correlation of 100 computer generated noisy barcodes of the R100 plasmid as a function of the noise parameter \( \alpha \) in eq. (5). The intersection with the blue line at cross correlation 0.9 is where the desired value of the \( \alpha \) parameter is, since experimental cross correlations for same-type plasmids tend to stay above this threshold.

## 8 Results

Each of the three different algorithms were run 100 times on the artificial insert dataset (see section 6.2) with different impositions of noise and the results were evaluated as true
(i.e. a successful run) or false (i.e. a failed run) according to the criteria presented in section 7. We found that the DTW algorithm had the best performance by finding the insert 93% of the time. The adapted LCMA segment detection method had the second best performance and found the insert 75% of the time while the MVM algorithm only found it 26% of the time. When the real path was shifted more to the right in the CM, the MVM performance dropped to 22% while the other performances remained unchanged. Figures 10-12 shows examples of both successful and unsuccessful runs for the three different algorithms.

**Figure 10:** Example of 2 out of 100 results when running the DTW algorithm 100 times on the same setup with varying phase-randomized noise in order to better mimic experimental consensus barcodes. The DTW path is plotted in red and the real path (exact match ignoring the addition of noise) is plotted in blue. The background color map is the CM. The horizontal barcode contains an artificial insert around Y-index 150 which can be seen as a gap in the blue path. In figure (A) the algorithm has found the insert and in figure (B) the path has clearly departed from the real path and therefore the algorithm was unable to find the insert.
Figure 11: Example of 2 out of 100 results when running the MVM algorithm 100 times on the same setup with varying phase-randomized noise in order to better mimic experimental consensus barcodes. The MVM path is plotted as red circles and the real path (exact match ignoring the addition of noise) is plotted in blue. The background color map is the CM. The horizontal barcode contains an artificial insert around Y-index 150 which can be seen as a gap in the blue path. In figure (A) the algorithm has found the insert, but did not make a particularly good match after the insert. In figure (B) the algorithm did not find the insert. This run clearly demonstrates the expected issue with the MVM method where the path has made an undesirable jump and is subsequently unable to return to the real path.
Figure 12: Example of 2 out of 100 results when running the adapted LCMA segment detection algorithm 100 times on the same setup with varying phase-randomized noise in order to better mimic experimental consensus barcodes. The LCMA path (see Figure 8) is plotted as red circles and the real path (exact match ignoring the addition of noise) is plotted in blue. The background color map is the CM. The horizontal barcode contains an artificial insert around Y-index 150 which can be seen as a gap in the blue path. In figure (A) the algorithm has found the insert. This run also demonstrates the need for outlier removal or a smoothing function since it is obvious that the lone matches are misclassified. In figure (B) the algorithm did not find the insert.

8.1 Dependence on Noise Level

The fraction of hits as a function of the noise parameter $\alpha$ for the different algorithms are presented in Figure 13 below. The fraction of hits for each value of $\alpha$ is averaged over 20 different runs. Clearly, the DTW is the most accurate of the algorithms when it comes to
this particular problem. However, it is possible that the problems discussed in section 4.1 becomes more apparent in other cases, especially, as mentioned, if the insert is longer. It should be noted that many of the miss pixels in the LCMA method are obvious outliers since many are lone pixels far away from an otherwise obvious apparent diagonal path (see Figure 12 A). Hence, a method for outlier detection/removal could be of very good use. From Figure 13 it seems that the DTW algorithm has a slightly steeper decline i.e. is slightly more vulnerable to noise than the other methods.

![Fraction of hits vs α parameter](image)

**Figure 13:** Fraction of hits (see section 8) as a function of the noise parameter $\alpha$ for the different algorithms discussed throughout the thesis. The fraction of hits for each value of $\alpha$ is averaged over 20 runs. A higher value of $\alpha$ equals more noise as described by eq. (5).

9 Conclusion and Outlook

Neither of the two new algorithms tested in this thesis managed to outperform the basic DTW algorithm. The MVM algorithm, as described in the result section, suffers from a slight dependency on the relative location of the correct path in the CM. This, in conjunction with an otherwise very bad overall score, renders the method virtually inapplicable to the barcode comparison problem (see Figure 11 B). The LCMA algorithm exhibited no clear faults, but only produced decent scores at best, subpar to the DTW.

Although the adapted LCMA segment detection method did not perform very well in terms of the results presented, I believe that it could potentially be refined to become a decent method for discovering not only inserts, but other SVs as well. Keep in mind that the method that was devised for defining a final one-to-one path was only an exploratory
attempt in order to better compare the method to MVM, and not a lot of thought was put behind it. Most likely, the valuable information in the LCMA segments could be used in other ways to construct a better path. So far, the method also seems to have no obvious drawback as both DTW and MVM has. The method is also highly flexible with its many parameters. Another thing that was not mentioned before is that the adapted LCMA segment detection method is the only one of the three methods that has the potential to detect an SV if the consensus barcode is "cut" in the middle of the SV. In such a case, the upper blue path of Figures 10-12 would be to the right of the lower part instead of the left.

Since the issue with the MVM algorithm was that it "jumped" left too early, and did not manage to return to the correct path again, it is easy to imagine that a possibility for the algorithm to jump right would fix this fatal problem. This is one thing that could be subject to future research. Another thing that could be of interest is exploring more of parameter space in the LCMA segment detection method. The number of parameters and the restricted time limit hindered our ability to delve too deep into this issue and instead we used reasoning to support our choices. Most notably this goes for the bandwidth parameter, but it is likely that other values of the bandwidth would affect the performance in the \( L-T \) landscape. It might also be interesting to investigate the different methods dependence on the length of the sequences and the insert.

An additional area of interest would be to investigate the effects of coarse-graining the methods. Instead of comparing the sequences pixel-by-pixel, it is possible to group the pixels up and compare them with their combined cost of for instance four pixels together. This method is used in different areas of computational biology and it is possible that it would improve the methods’ performance.
Appendix A: The Linear Assignment Problem Approach

The linear assignment problem is a classic problem in combinatorial optimization. In his article from *Journal of the Society for Industrial and Applied Mathematic* in 1957 [25], J. Munkers described it as follows:

"The personnel-assignment problem is the problem of choosing an optimal assignment of n men to n jobs assuming that numerical ratings are given for each man’s performance on each job. An optimal assignment is one which makes the sum of the men’s ratings for their assigned jobs a maximum."

It can be directly applied to the case of barcode alignment since the CM already holds ratings (score) for each man-job (X-pixel - Y-pixel) pair. Assuming that every job does not need to be assigned to someone, it becomes equivalent to partial shape matching of the barcodes. Here, we have used an implementation by Umut Orguner of Murty’s [26] classic algorithm for solving this problem, which also returns the $k \in \mathbb{N}$ best paths. Graphical plots of the assignments are shown below.

**Figure 14**: Assignment results for the Linear Assignment Problem approach. (A) When comparing two identical sequences, the algorithm makes a perfect match. (B) When introducing noise, the assignments become totally scattered and seemingly random. In this plot, the noise parameter $\alpha = 0.138$ is the same as for the other methods in the result section.

The above result is very strong indication that we cannot realistically use Murty’s algorithm to solve the problem of barcode shape matching (at least without proper coarse-graining). For real experimental sequences, there is always noise present.
Appendix B: The Phase Randomization Technique

The phase randomization technique was introduced by the Ambjörnsson group in order to create realistically looking semi-random barcodes. The reason for the requirement of such a technique is because purely random sequences (just a string of randomly generated, equally probable A, T, G or C characters) tend to render theory barcodes that have many more features than experimental barcodes of the same length. The most obvious reason for this is that DNA strings from real organisms typically have periods of very A-T or G-C rich regions, making the sequences not completely random [27]. To atone for this, the phase randomization technique utilizes Fourier transforms in order to randomize the phase of the new barcode.

For an input barcode $X$ of length $N$ the discrete Fourier transform is applied according to

$$\tilde{X}(k) = \sum_{n=1}^{N} X(n)W_j^{(n-1)(k-1)}$$

where $W_j = e^{\frac{2\pi i}{j}}$ is one of $j$ roots of unity. This Fourier transform is then multiplied with random phase factors $P_k$

$$P_k = e^{2i\pi \phi_k}$$

where $\phi \in [0,1]$ is a discrete even function of uniformly distributed random numbers. The new phase randomized barcode $R$ is then obtained from the inverse Fourier transform of the phase randomized $X(k) = X(k)P_k$, ie:

$$R = \frac{1}{N} \sum_{n=1}^{N} \tilde{X}(n)W_j^{-(n-1)(k-1)}$$

The evenness of $\phi$ ensures that the new barcode $R$ is real-valued.
10 References


