Contig assembly and plasmid analysis using DNA barcodes

Master thesis

The Department of Astronomy and Theoretical Physics

2016 – 01 – 15

Author
Christoffer Pichler

Supervisor
Tobias Ambjörnsson

LUND
UNIVERSITY

1. Abstract

Two methods of computational analysis of DNA barcodes are presented. A DNA barcode is formed by making GC-rich regions of a DNA molecule fluoresce while AT-rich regions remain dark, thus when stretched using nano-channels and viewed in a microscope, the DNA molecule will resemble a barcode with black and white stripes. Because of point-spread functions and pixellation the resolution will be roughly one data point per 200nm (or roughly 700 bp). This resolution is typically enough to distinguish between two different DNA molecules.

First DNA barcodes are used for analyzing an antibiotic resistance outbreak. In the outbreak, antibiotic resistant bacteria infected newborn children at Sahlgrenska University Hospital. The bacteria were of different strains and it was suspected that the bacteria shared the antibiotic resistant gene with bacteria not containing it through the exchange of plasmids. A plasmid is a short circular DNA molecule, typical length between 2 kbp to 1 Mbp (base pairs), which bacteria use to store genes that benefit survival (such as antibiotic resistance genes).

The second method is about matching short pieces of DNA sequence, called contigs, to a long intact barcode (from the same molecule as the contigs) to figure out the order of the pieces of sequence. In order to match a sequence to a barcode, the sequence has to be converted into a theoretical barcode first. After that it is compared to the long barcode, to find the optimal placement. Contigs are not supposed to overlap, and that is an assumption used in the methods presented in section 7.

The matching in both methods is facilitated by the use of our new statistical tools in order to reduce the number of false positives in the matching process. The results for the plasmid tracing method show that the method can be used to trace plasmid spread. On the other hand, the results for the contig assembly show that the method has potential to be useful, but at the moment it has been unsuccessful at assembling real contigs into a full, correct, sequence.

2. Purpose

There are two purposes of this thesis. Both purposes are related to developing theoretical and computational methods to facilitate time demanding steps for practical uses of DNA barcodes.

Bacteria are able to transfer genes between each other by exchanging plasmids. If one bacterium obtains an antibiotic resistance gene, then it can spread the gene to “nearby” bacteria (from another strain) and this can cause problems for humans. If either one of the strains is transferred to another human, the antibiotic resistance problem persists. Not everyone that becomes infected (from a certain disease) must have the antibiotic resistant strain, and by making a DNA barcode of the plasmid containing the antibiotic resistance gene, the spread can be traced. The first purpose of the thesis was to show that this tracing method was possible, using data obtained from an outbreak of ESBL-producing bacteria at Sahlgrenska University Hospital, seen in section 6.

When sequencing DNA, the molecule is divided into short pieces of sequenced DNA which are then assembled as much as possible, using the overlap between edges, until there are no overlap between them. These long sequences are called contigs. To assemble the contigs into a complete DNA sequence, the order of the contigs must be determined. For e.g. human DNA we know roughly how the DNA should look like, and can use that as a template. For a previously non-sequenced species (de novo sequencing), there is no template. By transforming the contigs and the complete DNA molecule into DNA barcodes, the order of the contigs can be determined. The second purpose of the thesis was to develop computational methods to accurately order the contigs using DNA barcodes, seen in section 7.

In both parts of the thesis, statistical tools were developed to quantify the quality of a match between two barcodes. In connection to this, effects such as pixellation and Point Spread Function are taken into account.

3. Introduction

Since its discovery in 1869, the study of DNA has come a long way. Today, we know that DNA molecules contain genetic information in all of the world's living organisms, from humans to bacteria. By studying the DNA sequence from a living organism, part of the properties of that organism can be predicted. For example, if there is a specific gene, coding for brown eyes, in a human's DNA molecule, then that human will have brown eyes. This information could potentially be extracted without ever seeing the person's eyes. The issue is how to extract the genetic information from a DNA molecule.

The DNA molecule is a double helix that consists of four bases, adenine (A), Thymine (T), Cytosine (C) and Guanine (G), and these are paired so that A is always opposite to a T in the double helix structure. The combination of A and T, or C and G, is called base pairs. These bases are impossible to see with our eyes since they are roughly 0.3 nm wide, and thus some sort of experimental technique is required to learn the order of the bases (or in other words the DNA sequence). Even using a microscope it would not be possible to see the base pairs because of the diffraction limit (which limits microscopy resolution to around 300 nm).

In this thesis, data from an outbreak of antibiotic resistant bacteria at Sahlgrenska University Hospital (2008) that is described in more detail in section 6.1, was studied. The outbreak was suspected to have occurred because the different strains of bacteria exchanged a plasmid containing an antibiotic resistance gene. The genetic information, the DNA, of a bacterium is stored in two ways. Most of it is stored in a large "package" known as a chromosome, but there are also smaller pieces of DNA that are not attached to the chromosome and these are called plasmids. In the chromosome all the genes that the bacterium always needs are stored, such as information about construction of the cell wall or ribosomes. On the other hand in the plasmids, things that are needed at that time are stored. An example of this is genes containing antibiotic resistance, if the bacterium is in an environment with antibiotics (e.g. inside a patient at a hospital). The plasmids are dynamic and change depending on the needs of the bacterium, and two bacteria can also exchange plasmids. Because of bacteria's ability to exchange plasmids, it is enough that one bacterium develops resistance to antibiotics and it can spread the resistance to the rest.

A plasmid is much shorter than the entire chromosome of the bacterium and is usually around 50 to 200 kbp (kilo base pairs). This means that it is possible to extract, and convert, intact plasmids into DNA barcodes. The barcodes from different bacteria can then be compared in order to track if a plasmid, that e.g. might contain antibiotics resistance, has been spread to other bacteria. The methods described in section 6, make the tracking of plasmid spread quick and possible.

"Sequencing-by-Synthesis" is a well known method from the late nineteen hundreds. Groups like M. Ronaghi et al.[1] and E. Kawashima et al.[2] have their own approach, but they share that a complimentary DNA strand is synthesised in order to sequence the DNA. They also have in common that plenty of samples have to be obtained and grown in order to finish the sequencing. This is possible to do and it has been done plenty of times, but there are benefits of using new techniques. Another drawback with this method is that the sample is destroyed by the process of analysing and in order to change something, a new sample has to be acquired.

During the new millennium, optical DNA analysis has grown rapidly. H. Parab et al.[3] used gold nanorods to detect and single out specific target DNA from a mix. This method does not actually

sequence anything, but is instead used to identify, “fingerprint”, if a DNA molecule is something previously encountered. The limitation here is also that only one target DNA can be used at a time, so the mix has to be tested many times in order to determine what DNA molecules that are in there.

A newer method, that is being worked on experimentally, both in Lund by Jonas Tegenfeldt’s group[4] and at Chalmers by Fredrik Westerlund’s group[5], is called DNA barcodes. The name comes from that parts of the DNA molecules are stained with a fluorescent molecule (YOYO-1) under conditions that allows it to bind sequence specific, and thus some parts will emit light while others are dark. Several images of DNA stretched in nanochannels are recorded and then stacked into a, so called, kymograph of the DNA molecule and the intensity vector of the mean of these images will resemble a barcode.

In order to process and compare the data between different barcodes, theoretical work is required. In this thesis it will be shown how DNA barcodes can be used as a final step in a sequencing process, as seen in section 7, and detecting if DNA molecules are the same, as seen in section 6. This method allows us to extract data from a DNA molecule once and then use this data any number of times without any degradation, unlike previously mentioned methods for DNA analysis. Since the intensity vector of the DNA molecule is saved on a computer, it can be used both for sequencing and for detecting similar DNA molecules from a mix of samples. Also, when new fine tuning discoveries are made, the software analysing the DNA barcodes can be changed and then all the old barcodes can be reanalyzed in order to stay relevant even though our knowledge of DNA expands.

4. Background information

The goal of the thesis is to compare DNA barcodes to each other and obtain useful results. To help make the comparisons useful, statistical tools are used in order to classify comparisons as matches, part-match, or different underlying molecules. To do all of that, background information about DNA barcodes is required. This section will describe how a barcode is created and the basic principle of how barcodes then are compared to each other.

The name DNA barcode comes from the similarity between the gray scale images of stained, and stretched out, DNA molecules and barcodes that we are familiar with from supermarkets, see Figure 1. Both types of barcodes are also created by humans to accomplish the same thing: Obtain a unique, and easily accessed, "fingerprint" for a specific DNA strain (or object in the everyday case).

4.1 Staining and stretching

The first step of creating a DNA barcode is to stain it in a base pair sensitive way. Jonas Tegenfeldt et al. are working with a method known as DNA Melting [4], while Fredrik Westerlund et al. are using the method Competitive Binding (CB)[6]. All results in this thesis originate from barcodes created through the CB approach.

The CB method uses two binding molecules. The first, YOYO-1, is a fluorescent dye which binds to every part of the molecule with no preference. In order to get distinction from areas with high concentration of AT or GC, another molecule is added called Netropsin. This molecule does not fluoresce, but it does have a preference to bind itself onto AT rich sites. By mixing both molecules with the DNA, in a perfect world, all GC rich areas would light up while the AT rich areas would remain dark. This is not completely true since the molecules bind with a certain probability and they are competing for binding sites, and thus the name: Competitive Binding.

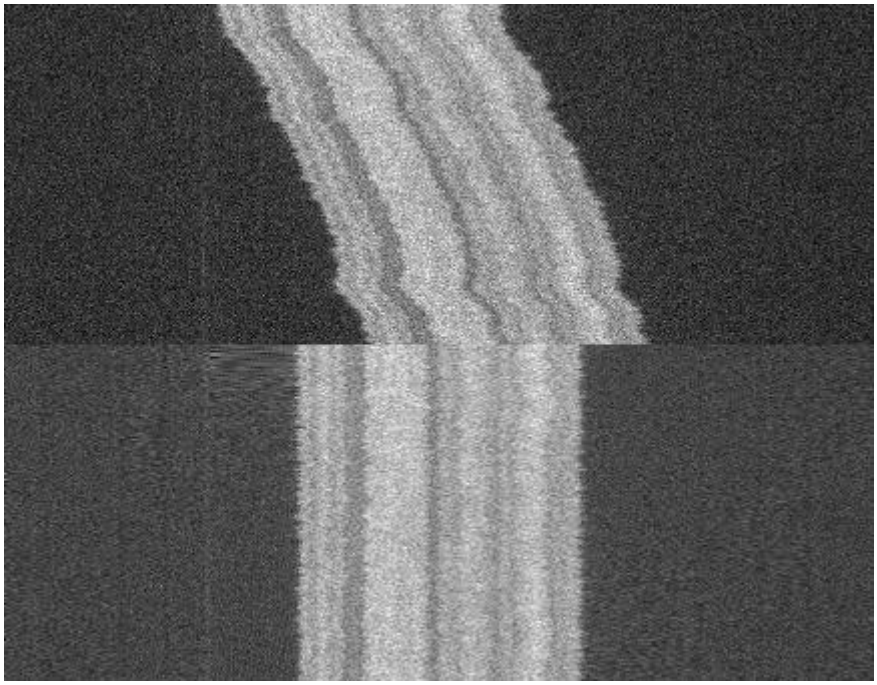


Figure 1. (Top) An unaligned (raw) barcode. Each row represents one frame of a video that recorded the intensity of the barcode inside a nano-channel. (Bottom) The aligned version of the top barcode. The algorithm looks for common features from each row and aligns them under each other. The time average, average along the columns, is what makes up the barcode intensity values used for matching purposes.

Once the DNA is stained in a useful way, independent of the method, it is introduced to small nano-channels so that the twirled and compact molecule is stretched out. If the DNA is circular, e.g. a plasmid of a bacterium, it is first cut to become linear. The molecule is cut using photocutting at a random point (spontaneous process while the molecule is exposed to light). The stretched DNA molecule is then photographed up to 100 times and the intensity in each pixel is recorded. The intensity variation along the nano-channel makes up all the information of the DNA barcode. An example of a single DNA barcode can be seen in Figure 1, both before and after aligning. Since the molecule is diffusing back and forth, an alignment algorithm[7] has to be run in order to get the aligned (and useful) version of the barcode. If the barcode was not aligned, the time average that is extracted from each column would not give any useful data.

4.2 Consensus barcode

When using a single experimental barcode there can be several issues that destroy information. One common problem is background noise. The experimental barcodes are photographed in nanochannels, but there is more than one molecule in the nanochannels simultaneously. Light from the other molecules as well as diffusion within the molecule (some areas spontaneously are stretched out while others are compressed) will both contribute to noise in each pixel, or variations from a mean intensity. Another issue is that the ends of each molecule will have less intensity, since the PSF (Point Spread Function) will mix the background intensity with the molecule's edge intensity (discussed more in section 4.3). A solution to these problems is to use more than one experimental barcode when forming the intensity vector to be used. A combination of several experimental barcodes is called a consensus barcode.

The consensus barcodes are formed by matching all experimental barcodes for a single plasmid (e.g. five barcodes) to each other[8]. The best match determines which two barcodes are being merged first. Within this matching process the best position for the two barcodes is found. Usually, the two barcodes have been cut at different locations (since the cut is made at a random location) and already after one merge, the combined barcode is better than a single one of them. The combined barcode is treated as any other barcode and the matching is done again. This is repeated until all barcodes have been merged into one consensus barcode. Note that when merging a combined barcode and a non-combined barcode, there are weights that will make the combined barcode contribute more to the mean (since it contains information of 2 or more barcodes).

This procedure solves the issues with the ends, since all the other barcodes should contain the missing information from the ends of one of the barcodes. It also reduces background noise, since this also is random fluctuations (but the mean should be the same in all the individual barcodes). For the rest of the thesis, experimental barcodes actually refer to consensus barcodes.

4.3 Point Spread Function

There are a few problems with the DNA barcodes. The most prominent issue is that the barcode do not contain information with base pair resolution; an example of a barcode (time average of the aligned kymograph in Figure 1) can be found in Figure 2. The resolution is limited by two factors. The first factor being that cameras use pixels to record intensity and, in this case, each pixel is 159.2 nm wide while a base pair is roughly 0.34 nm. The size difference alone means that each pixel will contain information from 470 base pairs. The second factor, which contributes even more, is what is commonly known as the diffraction-limited Point Spread Function (PSF).

The width, d, of the PSF is given by equation (1), and depends on the wavelength, λ, of the fluorescence light and the f-number, N, of the equipment.

$$d = 2.44 \cdot \lambda \cdot N \tag{1}$$

In our case, the width of the PSF is roughly 300 nm (experimentally determined), which is twice the size of a pixel, and it covers 900 base pairs. Because of the PSF the signal from the DNA molecule recorded in our barcode will be blurred, meaning that the intensity in one pixel is a weighted sum of the intensity from several hundred base pairs. Even with this limitation, the DNA barcodes have a sufficiently unique fingerprint for distinguishing most plasmids. An example of an experimental barcode can be seen in Figure 2. A schematic overview of the barcode generation process can be seen in Figure 3.

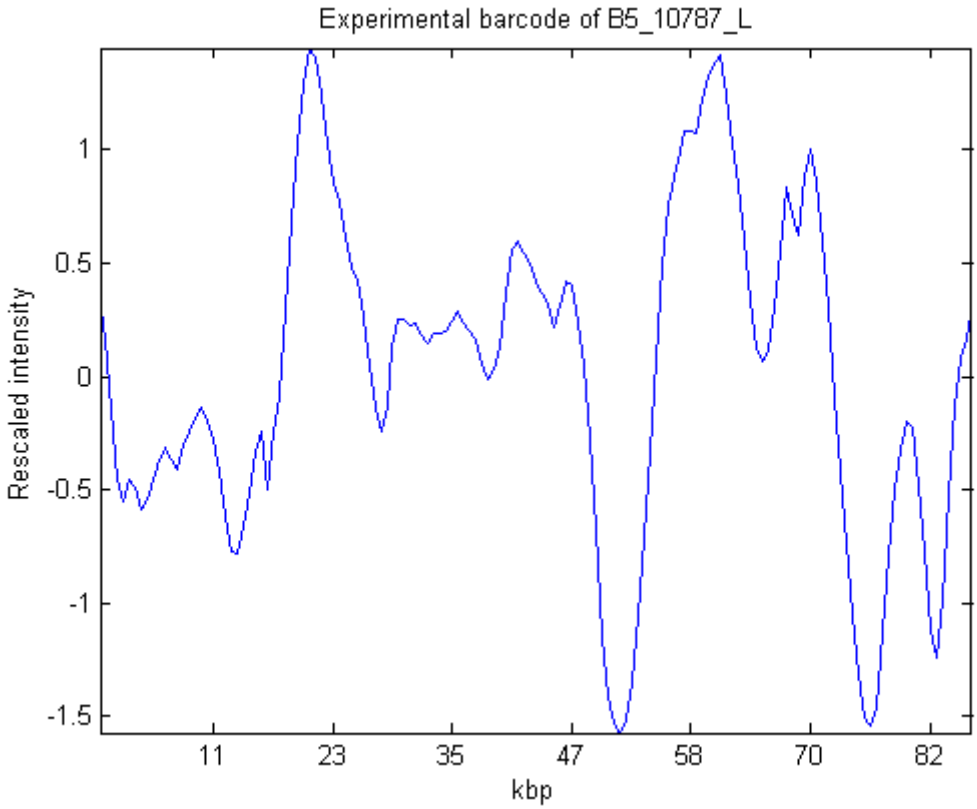


Figure 2. A rescaled (mean subtracted and scaled by standard deviation) barcode with intensity plotted against length along a nano-channel. The general features of the barcode have a width of at least 1 kbp.

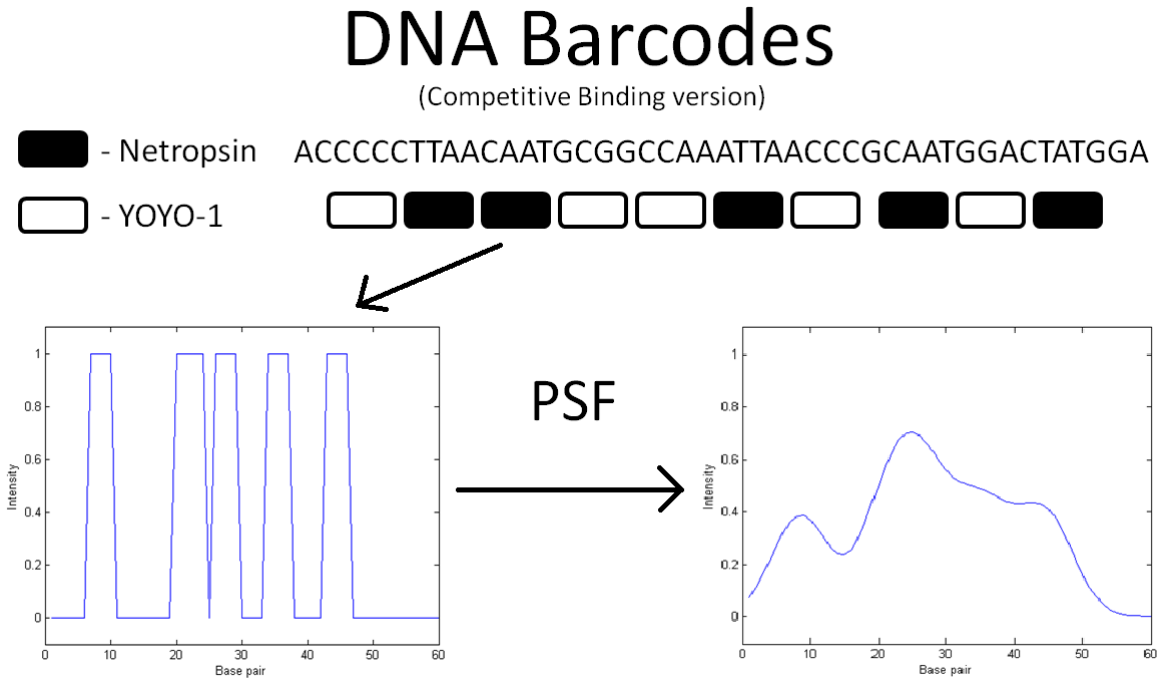


Figure 3. A schematic overview of the relation between sequence and barcode. Under the sequence there are boxes that represent the two staining molecules. YOYO-1 (white boxes) is assumed to have intensity equal to 1 while Netropsin (black boxes) is assumed to have intensity equal to 0. YOYO-1 can bind to any location, but Netropsin binds only to AT-rich regions and it binds stronger than Netropsin. After the first arrow, an image of how this barcode should look is found, but because of the Point Spread Function (PSF), the barcode actually looks like the image to the right (after the last arrow).

4.4 Theoretical barcode

In some cases, it is of interest to convert a known DNA sequence to a barcode; an example of this can be seen in section 7. Depending on which experimental approach the barcode is supposed to resemble, the computation is somewhat different, but only the CB approach will be discussed. The barcodes are generated in the same fashion as Adam Nilsson et al. do in their paper[5].

The intensity of the barcode from every base pair will be proportional to the probability that a YOYO molecule is bound to that base pair. The binding molecules are four base pairs long, which means that the probability in one point will depend on the neighbouring points. Calculating the probabilities in each point is done using statistical physics. The input parameters include binding constants for the two binding molecules (YOYO-1 and Netropsin) as well as concentration and size of the molecules.

Once the probabilities have been calculated, the two blurring effects have to be simulated. The PSF is simulated by convoluting a Gaussian with $\sigma = 300\ \mathrm{nm}$ with the probability function. Afterwards, the barcode is divided into bins, of the same size as a pixel, and the non-weighted mean intensity in each bin is calculated. These bins then form each the pixel values of the theoretical barcode. The process is similar to what is seen in Figure 3, except that the molecules positions are calculated and after the PSF step, there is also a pixellation step.

5. Quantifying barcode similarity

Since the barcodes are supposed to act as identification, it is important to be able to identify if two barcodes are the same or not. There are many applications for this kind of test, but this thesis

focuses on contig assembly, section 7, and bacteria tracing, section 6. It is impossible to say definitively that two barcodes are the same, but the probability that the cross correlation, equation 2, was high by coincidence can be calculated and is represented by a quantity called p-value.

5.1 Cross correlation

To quantize the resemblance of two barcodes, a number between -1 and 1 is calculated. This number is called the cross correlation, *C*, between two barcodes and is calculated using equation (2). To have a normalized scale for the cross correlation values, all barcodes undergo Reisner Rescaling before comparison. The rescaling subtracts the mean and divide by the standard deviation of the intensity vector, thus resulting in intensity values with mean equal to zero and standard deviation equal to one.

$$C = \frac{1}{n-1} \cdot \sum_{i=1}^{n} I_1(i) \cdot I_2(i) \tag{2}$$

Where $I_1(i)$ and $I_2(i)$ are the rescaled intensity values in point i for barcode 1 and barcode 2 respectively. When calculated in this fashion, a cross correlation of 1 means that the barcodes are identical, or perfectly correlated, and -1 is then the opposite, perfect anti-correlated. An example of cross correlation of 1 is if $I_1 = I_2 = \sin x$. Another example, but this time for cross correlation equal to -1, is if $I_1 = \sin x$, but $I_2 = \sin(x + \pi)$. For the final example, let $I_1 = \sin x$ and $I_2 = \sin\left(x + \frac{\pi}{2}\right)$, then the cross correlation would be equal to 0, which means that the two barcodes are uncorrelated.

There could be cases when the two barcodes are not the same length. Then the longer barcode will be cut to the same size as the shorter. In order to cover all possibilities for a potential match, this is done at every possible start position.

5.2 Zero Model

Even though the cross correlation scale is normalized, it is still not enough to only have a cross correlation value in order to determine if two barcodes are the same. Since there are information loss from blurring and pixilation, two barcodes that originates from two different DNA sequences could, theoretically, have the same intensity vector. In a more realistic case the two barcodes will not be identical, but may still share features that can be found in every barcode and thus affect the cross correlation value. Statistics can be used in order to set a lower threshold for cross correlation values that are considered to indicate that two barcodes are the same.

In order to calculate a p-value, the probability that the cross correlation between two barcodes was high by coincidence, a probability density function (PDF), $\Phi(C)$, of cross correlation values from random barcodes is required. This distribution is referred to as the Zero Model (ZM). Using equation (3), a p-value, P, can then be calculated.

$$P = \int_{C}^{\infty} \Phi(C')\, dC' \tag{3}$$

The Zero Model, if constructed correctly, ensures that the p-values are distributed uniformly between zero and one when comparing many pairs of random barcodes. Because of this, the p-value can, very loosely, be interpreted as the probability that the cross correlation value was high by

coincidence (since two similar barcodes should have a high cross-correlation value, and thus a low p-value).

Assume that two barcodes (barcode 1 and barcode 2) are going to be compared. The DNA molecules, which barcode 1 and 2 originates from, could be circular. Then it could be the case that they are not cut at the same point and thus will not match even though they might be the same. To solve this problem, one barcode is circularly permutated compared to the other and a number of cross correlation values is obtained and the largest cross correlation value, or the extreme value, is saved and considered to be the "correct" value for the comparison between the two barcodes.

To be able to calculate a p-value, a mathematical model of the Zero Model (or the extreme value distribution) needs to be used. The model used is a Gumbel PDF[9] with two unknown fitting parameters. The Gumbel distribution is an extreme value distribution and it was chosen since we are modelling the distribution of the best cross correlation value from a distribution generated when sliding one barcode across another. Other alternatives have been explored and one of them can be found in Appendix A. The PDF of the Gumbel distribution is found in equation 4.

$$\Phi(C) = \frac{1}{\beta} e^{-(z+e^{-z})} \tag{4}$$

Where $z = \frac{C-\kappa}{\beta}$. The two parameters β and κ are related to the mean, μ and the variance, σ^2 of the distribution, found in equation 5 and 6 (γ is Euler's constant). Both the fitting parameters (κ and β) are determined by moment matching. Moment matching uses the sample estimators for the mean and variance from the collected cross correlation values as the μ and σ^2 for the PDF. Using these values, a mathematical Zero Model can be obtained for that particular pair of barcodes.

$$\kappa = \mu + \beta\gamma \tag{5}$$

$$\beta = \frac{\sigma\sqrt{6}}{\pi} \tag{6}$$

Figure 4 shows an example of a Gumbel fit to experimental data using the PR method, discussed in section 5.3, to obtain "random" barcodes. The number of random barcodes used was 1000 which is sufficient to obtain reliable results.

The method can also be used for barcodes from linear DNA molecules, as can be seen in section 7.

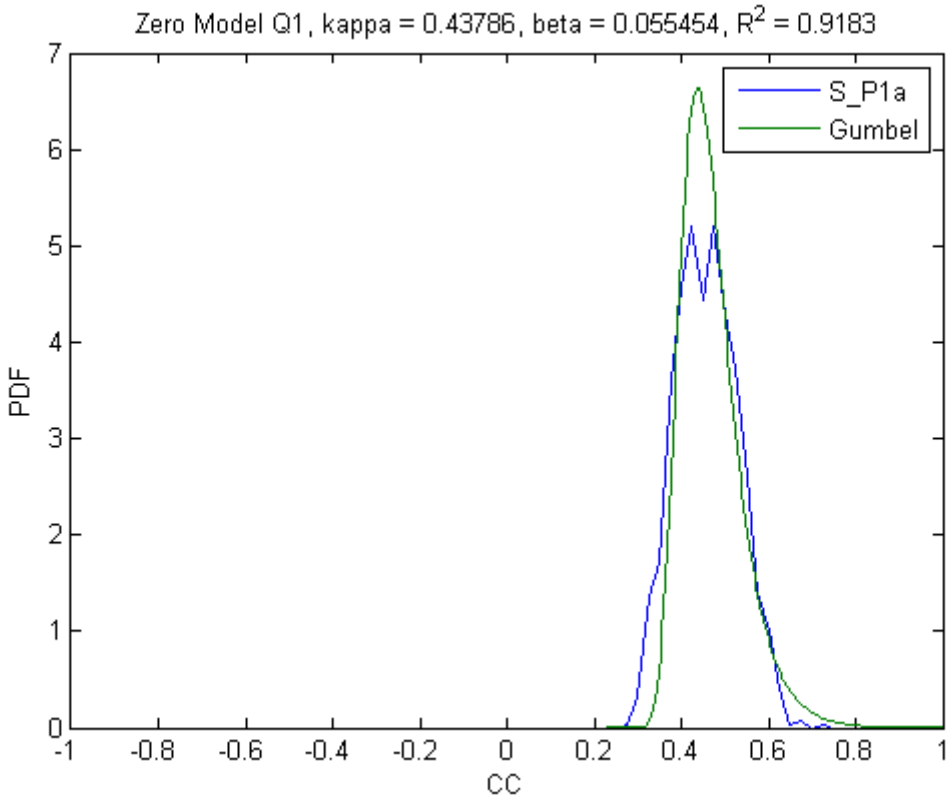


Figure 4. A Gumbel fit to a histogram of the largest cross correlation values from 1000 PR barcodes and an experimental barcode. A Gumbel PDF is defined by the value of $\boldsymbol{\kappa}$ and $\boldsymbol{\beta}$. The experimental barcode used was from a bacterial plasmid found in an outbreak at Sahlgrenska University Hospital, and is referred to as S_P1a. $\boldsymbol{R^2}$ is the coefficient of determination.

As previously mentioned, the p-values should be uniformly distributed between 0 and 1 if the experimental barcode is compared to another barcode that is not from the same sequence, e.g. a random barcode. Since the PDF of the cross correlation value from random barcodes is the same as a histogram of the cross correlations when enough barcodes are used, the Zero Model constructed with this method fulfils the uniformly distributed property of the p-value if the random barcodes used are representative of a random experimental barcode. If two barcodes are similar, then the cross correlation value between them should be high (which corresponds to a low p-value). That means that all barcodes that might be matching each other will have a p-value close to 0. By using a p-value threshold that is low enough, false positives can be sorted out while only keeping the actual matches. The value of the threshold cannot be too low, because then not even matches will pass it, so there are some standard values such as 0.05 or 0.01 (the latter is used when comparing barcodes from a real world problem in section 6).

5.3 Phase randomization

To generate the Zero Model (ZM) discussed in section 5.2, thousands of “random” barcodes have to be used. The citation marks are being used because the barcodes cannot be fully random, since no real barcode would look like that due to blurring effects, which correlates the data, and possibly some unknown intrinsic effects. The optimal “random” barcode would be another experimental barcode that is definitively not the same as the one being examined. Finding thousands of these might not be possible and even if it is, the daunting task of determining which barcodes that

represents a not to similar barcode would bring in subjectivity into the measurement. Instead of using barcodes directly, the method called Phase Randomization (PR), similar to what T. Schreiber et al. do in their paper[10], can be used. Introducing and adapting the PR technique to the DNA barcoding community is one of the main new developments in this thesis.

The first step in the PR method is to create Fourier Transforms (FTs) of all the chosen barcodes that is not the ones being examined. These barcodes are referred to as ZM barcodes; the ZM barcodes used to generate the FT in Figure 5 are Competitive Binding theoretical barcodes from all (3224) sequenced plasmids from Lena Nyberg et al.'s paper[8]. All the FTs have to be of the same length and since the length depends on the length of the ZM barcodes, the FTs are linearly interpolated to be the same length as the longest one. The barcodes cannot be interpolated directly because that would destroy information about the size of their features, and thus the FT has to be interpolated instead. It is also important that the FTs remain symmetrical after interpolation, otherwise the barcodes extracted later will be complex valued, and thus the FTs are only interpolated from one end to the middle and then they are mirrored. After the symmetrising process, a mean, of all the FTs, is formed and this FT serves as a general FT of a barcode (example can be seen in Figure 5). By multiplying, symmetrically, random phase factors to each frequency, a new FT is created. The inverse Fourier Transform of this new FT will be a general random barcode. By repeating the last two steps N times, N general random barcodes are generated and these can be used to calculate the Zero Model.

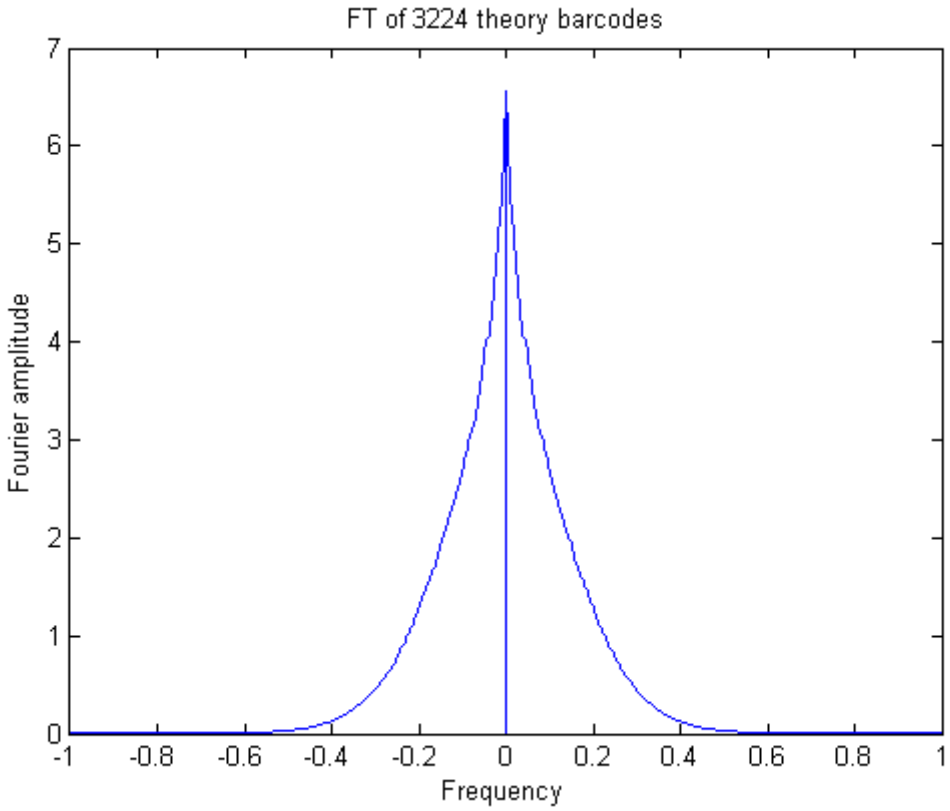


Figure 5. The mean of 3224 Fourier Transformed barcodes, that were stretched to the same length. The zero-frequency is zero because the barcodes are Reisner rescaled, thus having mean equal to zero (and standard deviation equal to one). An inverse Fourier Transform generates a "general" barcode. If random phase factors are symmetrically multiplied to the Fourier Transformed barcode, "random" barcodes are obtained when inversing the Fourier Transform.

In Figure 6, there are two barcodes generated using PR. They look clearly distinguishable, but still containing general features which make them resemble each other as well as the experimental barcode found in Figure 2.

Another way of generating random barcodes would be to generate a random DNA sequences and then make theoretical barcodes out of that. Assuming that actual DNA sequences behave like a random sequence (no preferences of neighbours), the computational cost is much lower for the PR method. Measured with an average computer (not state of the art, but not very old), generating 1000 barcodes using PR takes roughly 20 seconds. On the other hand, generating 1000 barcodes from random sequences takes roughly 2700 seconds. This process has to be done once for every pair of barcodes (if each barcode has different length), or once per barcode if all barcodes it is compared against have the same length. Using PR speeds things up with at least a factor 100. The PR method uses interpolation, in order to make the barcodes have the correct size and kbp/pixel, and FFT once per barcode. The other method first has to calculate probabilities by taking into account (usually up to some hundred thousand) neighbour interactions. After that it needs to calculate a convolution (using FFT) and lastly the quick pixellation effect consisting of roughly 150 averages. All of these steps need to be done for each barcode. Because of the speed advantage, as well as the fact that it could represent barcodes even better than random sequences (same autocorrelation function as the "input" barcodes), PR is used to generate random barcodes for the Zero Model.

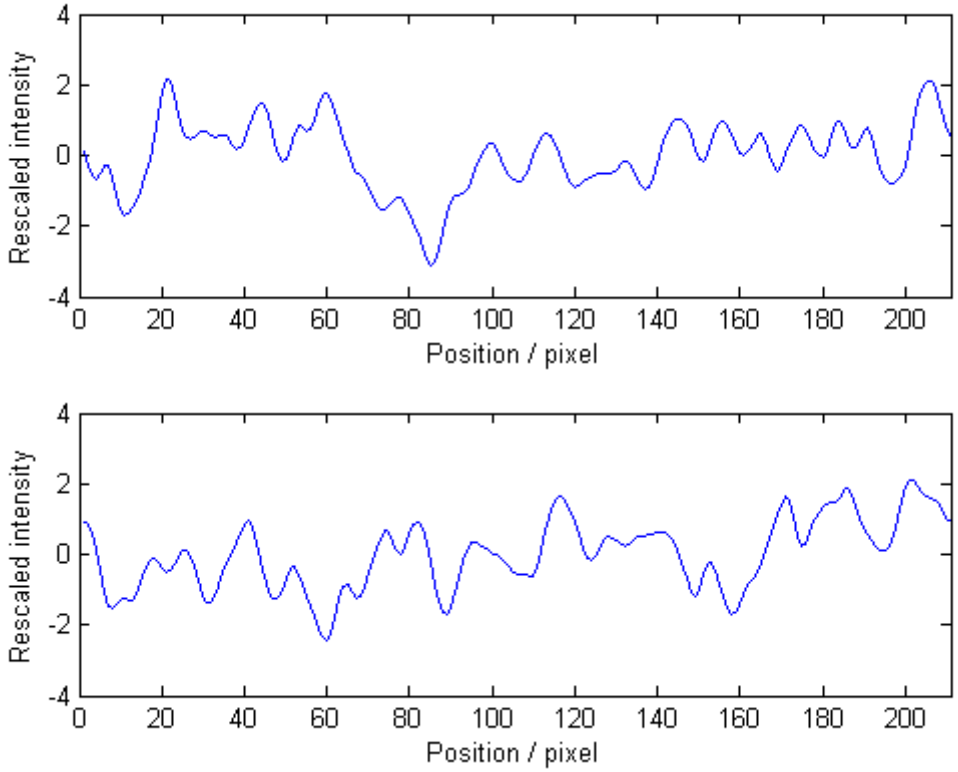


Figure 6. Two random barcodes generated using PR from the general FT barcode found in Figure 5. These can be used to generate Zero Models. They share similar features with the barcode found in Figure 2, but are still random enough to be able to represent another experimental barcode that is not related to any other.

In the paper[5] by Adam et al., an alternative Zero Model is presented. The main difference between these two is the way the “random” barcodes are generated. Since the random barcodes should be not related to experimental barcode that is being examined, while the general features of a barcode (such as PSF) have to be preserved, there are few options. Instead of finding the general Fourier Transform of barcodes and phase randomize it, random sequences could be made into (theoretical) barcodes instead. This could work really well if there is not enough data to construct a general Fourier Transform of barcodes, but there are also drawbacks. One of the drawbacks is that it has a higher computational cost to calculate the binding probability for YOYO-1 and Netropsin and then also convolute the barcode to simulate PSF, than just to add random phases and then inverse fast Fourier Transform a barcode. Another prominent drawback (that this method somewhat shares with the Phase Randomized barcodes) is that it is still theoretical barcodes that are used. That means that experimental noise and similar will not be there, and thus the random barcode will be less similar to an actual random DNA molecule that has been turned into a DNA barcode experimentally.

6. Plasmid mediated outbreak at Sahlgrenska University Hospital

An interesting problem, that can be solved using DNA barcodes and all the methods in section 4 and 5, is to look at gene transfer between bacteria, or monitoring plasmids during a disease outbreak. If several patients at a hospital suddenly become more ill than before, some disease might be spreading from patient to patient. The first course of action might be to put all of them into isolation and after that try and determine if they are all infected by the same bacteria. This might take days, or even weeks, before the lab have confirmed this. A quicker way would be to collect samples from all patients and then make barcodes of the plasmids found in the bacteria. Using the software developed alongside this thesis, all the barcodes can be compared and sorted into groups.

6.1 Outbreak

There was an outbreak of Enterobacteriaceae at Sahlgrenska University Hospital involved four patients that were 1-19 months old (median 3 months) at the neonatal post-surgery ward. Patient one (P1) and Patient two (P2) both had ESBL-producing (antibiotic resistant) E.coli, and Patient three (P3) and Patient four (P4) both had ESBL-producing K.pneumonia in addition to ESBL-producing E.coli. Both blood and faeces samples were collected from all the patients during the outbreak and additional samples were taken from P1 and P2, 5 and 17 months later respectively (still containing E.coli).

From each sample, the plasmids were extracted and labelled either “S_Px” or “L_Px” (for Patient X), depending on length. In each of the samples a plasmid of similar length was found and thus the name “S_Px” (“S” as in Similar). There were also a longer one in most of the samples, but there were no common length between these (and was thus not investigated). Several samples were taken from the same patient and those are distinguished by the last letter (“a”, “b” or “c”). Each of the similar (S_Px) DNA barcodes can be seen in Figure 7, after suitable pair wise “shifting” of the barcodes for finding the optimal positions (DNA are cut at random positions, see section 4.1).

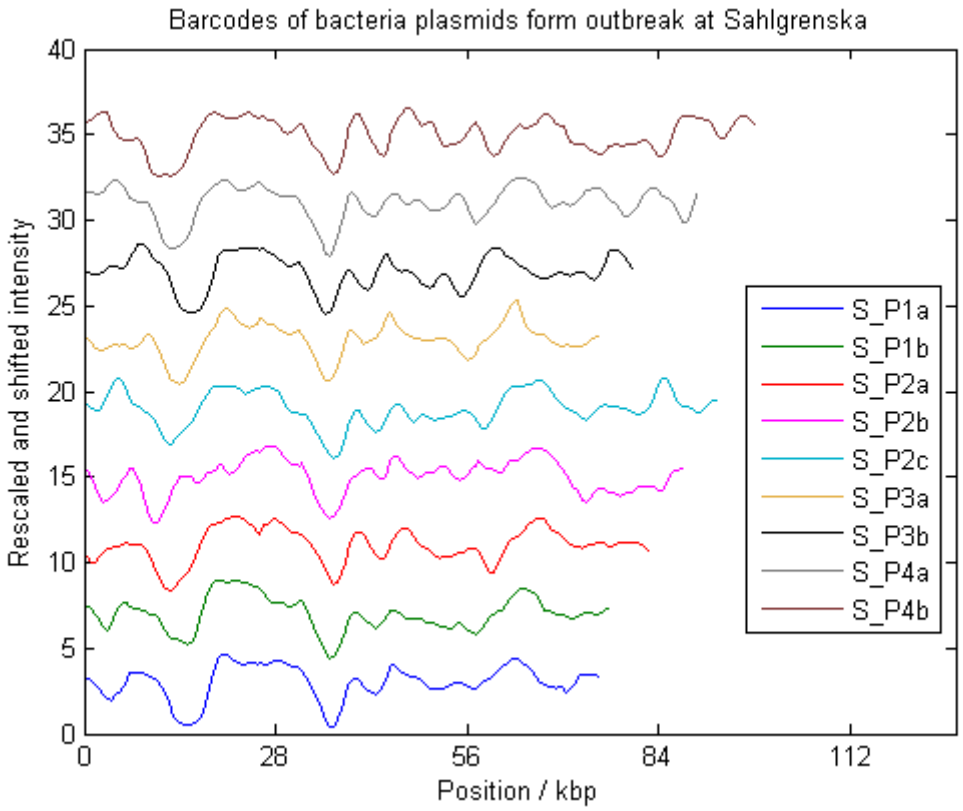


Figure 7. Shifted barcodes of plasmids found in four patients from an ESBL-producing bacteria outbreak at Sahlgrenska University Hospital. The number in the name corresponds to which patient the sample was taken from (e.g. S_P1a and S_P1b are from the same patient, but taken at different times).

6.2 Barcode comparison method

The software developed alongside the comparison method is called Experiment to Experiment (ETE), since it compares experimental barcodes to other experimental barcodes. There is no actual limitation for the software to only compare experiments, but the collaborators with this project are currently only using the software for that purpose. The software is used to answer one of the following two questions: “are these barcodes the same?” (Q1) or “is any barcode a part of another barcode?” (Q2). Which question that is being answered is determined at the start of the software since it affects some aspects of the comparison.

Why both of these questions may be of interest is that plasmids are dynamic and change content in order to suit the needs of the bacterium in its current location. So if a plasmid has been transferred from one bacterium to another and the new bacterium has moved, there might be one inserted gene that e.g. helps the bacterium survive in acidic environments (see Figure 7 for examples of nine different samples that originates from the same plasmid). Then the two barcodes would not be the same when stretched to the same length, even though the plasmids were the same before the bacterium adapted to the new environment. In order to accurately trace (e.g. ESBL-producing) bacteria spread, the plasmid barcodes must be allowed to have some inserted genes. Q1 is also interesting because it is much quicker and allows less overfitting, thus the results will be more distinct if it is a match or not.

To describe the software (and the method) and what it does in a clear and logical way, a summary list is presented below.

1. Input (experimental) barcodes.
2. Input ZM barcodes.
3. Stretch, by interpolation, all barcodes to the same length or kbp/pixel-value.
4. Generate a ZM for each pair of (experimental) barcodes.
5. Calculate cross correlation between each (experimental) barcode.
6. Use the ZMs to convert cross correlation into p-values.

The first two steps are straight forward. First the barcodes that are supposed to be compared to one another are inputted and this is followed by input of barcodes that will contribute to the general FT barcode.

Step 3 is the first that separates the two methods (Q1 and Q2). If one care only which barcodes are the exact same, then all barcodes should be stretched to the same length (otherwise they cannot be the same). If, on the other hand, it is suspected that some barcode might have an inserted gene, then all barcodes should be stretched to the same kbp/pixel-value. Since not all experiments are conducted with exactly the same setting, the stretching of DNA molecules can be different, e.g. depending on salt concentration, from barcode to barcode. Since the software only can compare pixel to pixel between two barcodes, it is important to have the same kbp/pixel-value in order to get accurate results. Otherwise, two identical sequences would not match very well since one would be long and dragged out while the other one would be short and compressed.

The last three steps are covered in section 5. The p-values are placed in a matrix in order to visualize it; an example can be seen in Figure 8.

6.3 Same plasmid Q1

As previously discussed, if two DNA barcodes are the same, they must have the same length. In Figure 8, an example of Q1 for ETE is seen (using the barcodes presented in Figure 7). There nine barcodes are stretched to the same length and then compared. It can be seen that there seem to be two groups of barcodes that fit well together. The groups are barcode number 1, 4, 5 and 9 and barcode number 2, 3, 6, 7 and 8. To visualize how well they match, examples of comparison between two barcodes of the same length can be seen in Figure 9 and Figure 10.

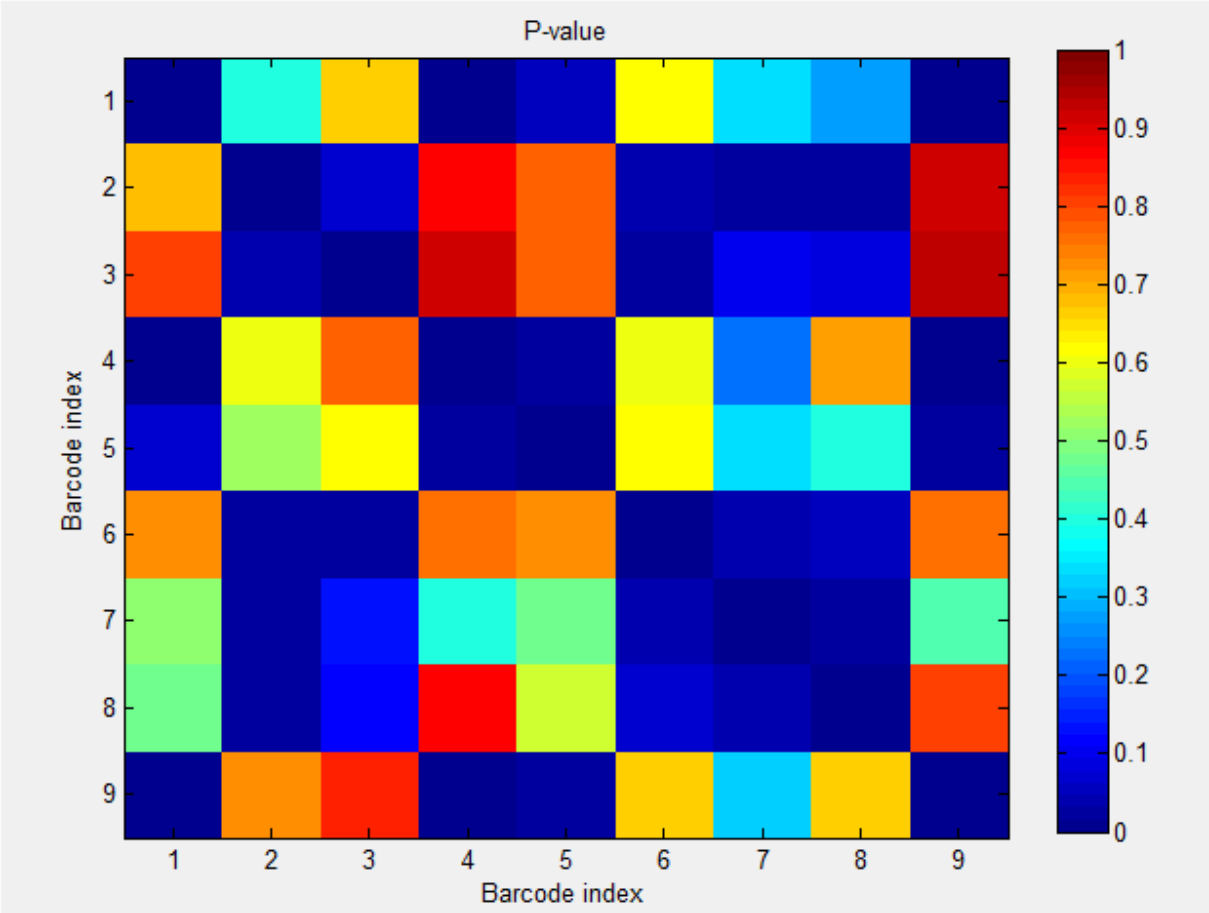


Figure 8. A p-value matrix after comparison of nine barcodes (found in Figure 7) using the method ETE Q1. From the p-values it can be seen that barcode 1, 4, 5 and 9 matches well together and that 2, 3, 6, 7 and 8 is another group of similar barcodes. This is determined by looking barcodes that have a p-value less than 0.1 when compared to another. The value 0.1 is chosen for this example just to demonstrate the division into groups, for the actual matching 0.01 is chosen instead. The nine barcodes corresponds to the nine barcodes found in Figure 7 (S_P1a is barcode 1, S_P1b, is barcode 2, etc).

In Figure 9, showing barcode 1 and 4 from Figure 8, the barcodes look very similar which is indicated by the low p-value, as discussed in section 5.2. In Figure 10, showing barcode 2 and 4 from Figure 8, on the other hand, the barcodes do not look similar which is indicated by the high p-value, as also discussed in section 5.2. The main issue in Figure 10, seems to be that barcode 2 (green line) is a bit stretched in the middle. This could be because they are not supposed to be the same length, because a new gene has been inserted into barcode 4. If barcode 2 is not forced to be the same length as barcode 4, and both of them are circularly permuted, then barcode 2 could, potentially, be found to be a part of barcode 4.

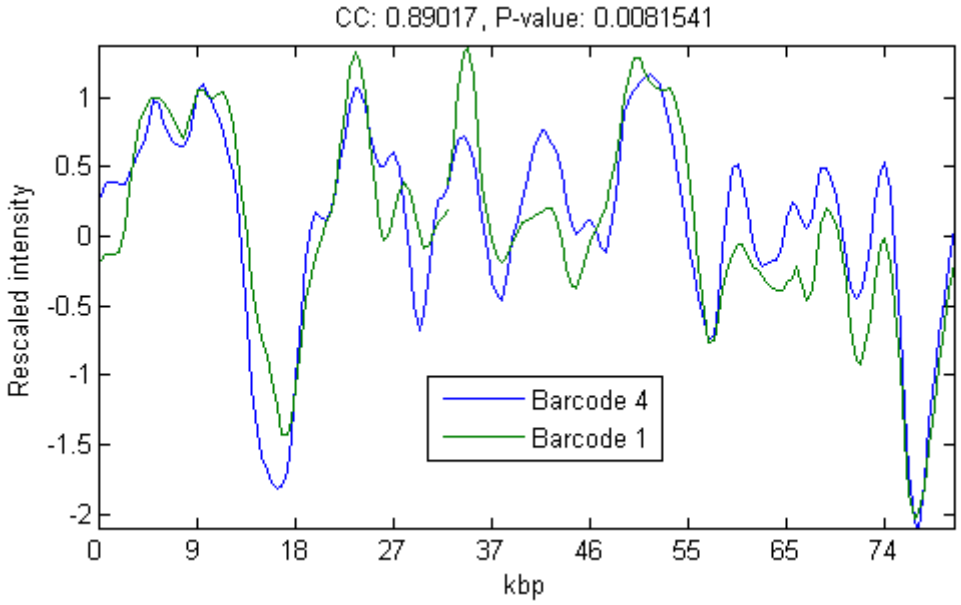


Figure 9. A good match between two equally long experimental barcodes (same as in Figure 8), matched with ETE Q1. The match is classified as good since the p-value < 0.01. It can also be seen visually that the two barcodes are very similar.

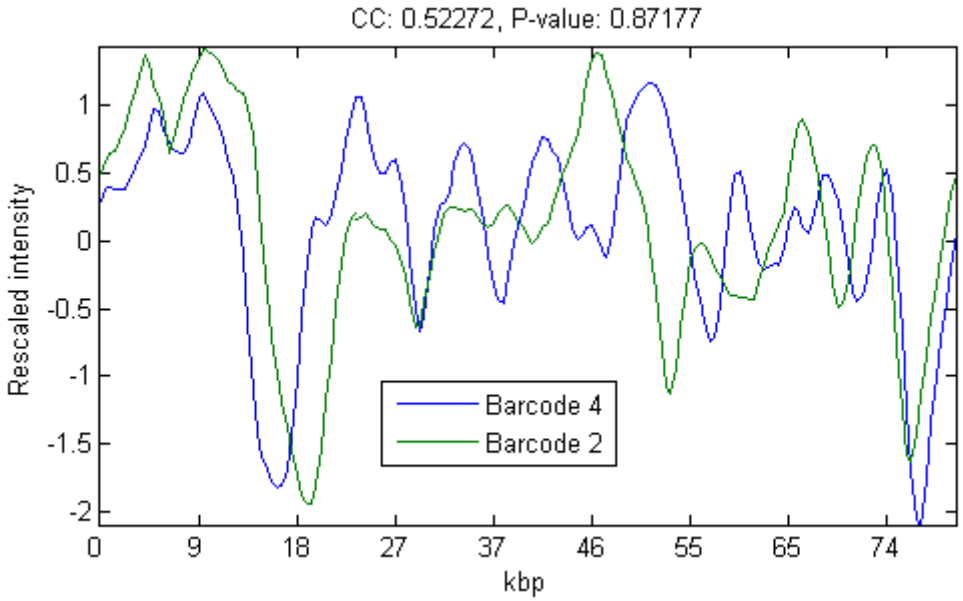


Figure 10. A poor match between two equally long experimental barcodes (same as in Figure 8), matched with ETE Q1. The match is classified as poor since the p-value > 0.01. This can be confirmed by visual inspection, since the two barcodes to not look similar everywhere.

6.4 Same plasmids with insert Q2

The main difference between Q1 and Q2, is that in Q1 only one of the barcodes are circularly permuted. A quick example to visualize this can be constructed by assuming the following. Barcode 1 is ABC while barcode 2 is BCA. By circularly permuting barcode 2, we get three possibilities: BCA, CAB and ABC. Matching barcode 1 with all of these permutations results in a matching being found (since

they, in this example, is the same barcode). If barcode 1 now has an inserted gene, it might be AXBC, where X is the inserted gene. All of a sudden, BCA, CAB and ABC are compared either to AXB or XBC and in neither case will there be any match. This effect is amplified if the barcode also is stretched to the same length where none of the new "pixels" (stretched pixel values) would have neither A, B or C as intensity value. The result would be that the two barcodes are not the same, which is true since barcode 1 has an extra gene.

To be able to find if there might be an extra gene in the mix, both barcodes has to be permuted. This operation divides barcode 1 into four barcodes, AXBC, XBCA, BCAX and CAXB, while barcode 2 still is BCA, CAB and ABC. With all permutations calculated, similar to how the two barcodes of different length were compared previous each version of barcode 1 will be divided into two possible places for barcode 2 to match. The second version would be XBC and BCA, and there can all of a sudden a match be found, showing that barcode 2 is indeed barcode 1 before gene insertion.

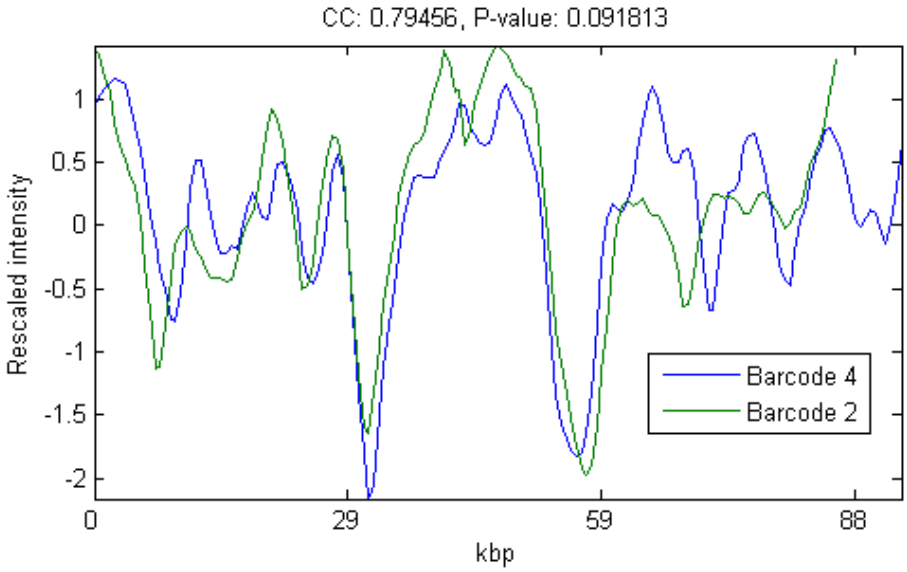


Figure 11. An almost good match between two experimental barcodes (same as in Figure 8) using ETE Q2. Barcode 2 is shorter than barcode 4 and both of them have been circularly permuted in order to find the match. The match is still not classified as good, but it is much better when one are of inserts were allowed in barcode 4 compared to the match in Figure 10.

In Figure 11, looking back at the previous example with barcode 2 and 4 from Figure 8, a better match can be seen than in Figure 10. There are still some areas that still do not match up, but most features are identified. A problem with the method in Q2 is that if there are two inserted genes at different locations, then the software cannot identify both of them. This problem might explain why some of the features seem to be shifted, while others are aligned. The general principle works anyway, and barcode 2 is found to be a part of barcode 4.

7. Contig Assembly

The traditional way of sequencing DNA starts with breaking the DNA into small pieces, sequencing each piece, and then matching them together using overlapping parts. When there are no more overlapping parts and only a couple of larger sequences remain, the process is stopped. The sequences that do no overlap with each other are called contigs. One way of constructing contigs is using a software called CAP that X. Haung et al. have published[11]. In order to obtain the full sequence the contigs must also be placed in the correct order. This contig assembly is a general problem for traditional DNA sequencing methods. The problem can be solved by matching short theoretical barcodes to a long experimental barcode.

When developing a method for the contig assembly, three different approaches (described in section 7.2, 7.3 and 7.4 respectively) were discussed. The first one, the simplified approach, is included only as a basic demonstration of the principles used within the two more sophisticated methods. Both methods has been used, at least when a p-value threshold has been applied to the Free Energy method.

7.1 Contig Preparation

The first, computational, step in any of the methods is to convert contig sequences to theoretical barcodes, see section 4.4. To do this in a proper way, the contigs need to have the same properties as the experimental barcode that they are going to be matched against. The kbp/pixel-value has to be the same; otherwise the barcode would seem compressed. Also, the width of the PSF has to be the same as for the experiment. These two parameters should be collected from the experimental setup. The theoretical problem that has to be solved is that the edges will behave strangely.

The edges will always have lower intensity values because of the PSF. Since the PSF basically spreads out the light from a point to a circle around that point, each point consist of light originating from several neighbouring points. At the edges on the other hand, there are no neighbouring points at one of the two sides. The lack of neighbours makes the intensity from the edges to be less than any other point in the barcode. This effect would not be a problem if the theoretical barcode being produced covered the whole experimental barcode, because then the edges of both the experimental and the theoretical would have the same fall in intensity, but since the theoretical barcode is just a small part of the experimental barcode, then the theoretical barcodes edges will miss some intensity from the neighbours which are not there.

The solution to the edge problem is to simply remove a couple of pixels on each side of every barcode. The phrase “a couple of pixels” refers to the interval of two to six pixels on each side, and it is written in this way because there is no objectively best way of removing the edge. The standard deviation of the PSF is roughly two pixels, thus 68 % of the problem would be removed by removing two pixels. If, on the other hand, six pixels were removed, 99 % of the PSF problem would be removed. The drawback with removing six pixels on each side is that the contig loses information from twelve pixels. Contigs are usually short, say 30 kbp or 50 pixels, which means cutting twelve pixels would mean losing at least 24 % of the information. The number of pixels to cut from the edges becomes a trade-off problem and can be changed between two matching attempts to see if the result changes.

7.2 Simplified Approach

This method, the simplified approach, has some problems with the results which will be discussed at the end of this subsection, but it is a simple model of how the other two methods work. The idea for the contig assembly is that each contig is one part of a jigsaw puzzle and an experimental barcode of the entire DNA molecule is the picture on the box, which is used to make the assembly process possible. By converting the contig sequences into theoretical barcodes and matching them to the long experimental barcode, the puzzle should be solvable without the contigs having any overlap. An assumption, that is supposed to always be true by definition of contig but is not guaranteed, is that the contigs cannot overlap with this method, not even by one pixel. A schematic overview can be found in Figure 12.

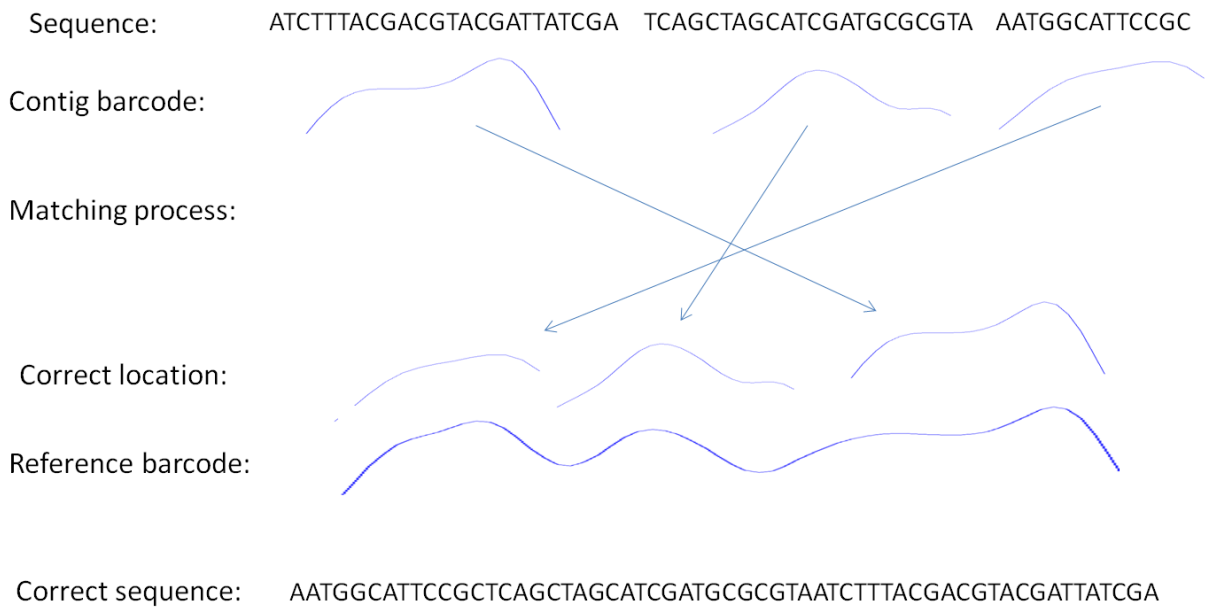


Figure 12. Schematic overview of the contig assembly idea. At the start there are only a reference barcode and contig sequences. The sequences are then transformed into DNA barcodes and then matched against the reference barcode in order to find the best position. When all contigs have been matched, their underlying sequence can be glued together to form the full sequence of the molecule.

A process list for the Contig Assembly (CA) software is presented below.

1. Input experimental barcode for the entire DNA molecule.
2. Input of contig sequences.
3. Input of ZM barcodes.
4. Conversion from sequence into theoretical barcode.
5. Matching contig lowest p-value contig to experimental barcode.
6. Checking if any of the other contigs are overlapping.

First the files used are read into the software. After that, the contig sequences are converted into theoretical barcodes, as discussed in section 4.4. Then, each contig is compared to the experimental barcode. This step gives all the contigs a p-value and a location on the experimental barcode. The contig with the lowest p-value is placed first. If the contig with second lowest p-value does not overlap with the first one, it is also placed. Otherwise it is matched again against the remaining parts of the experimental barcode, and then it is placed on the new location with lowest p-value. This

process is repeated until all contigs are placed. The result of a matching process like this for R100 can be seen in Figure 14. What is seen there is actually a branch of the Tree method, but it is done in the same way as the simplified approach.

As previously mentioned, there are some issues with this method. The first issue is that the contigs are not allowed to overlap. This could be an issue if some contigs have been assembled incorrectly and still have some overlap or if there are some problem with some region of the experimental barcode so that the ends of two contigs should be on the same pixel. These are minor issues and should not be a problem if the experiments and the making of contigs are both done perfectly.

The second, and most prominent, is that the lowest p-value contig is placed on the long experimental barcode first and then it can never be moved. In theory, this sounds as the best alternative, but the p-values should not be interpreted as flawless objective truths; after all, the p-value can only be loosely interpreted as the probability that the cross correlation between the contig and a piece of the experimental barcode was large by coincidence. Since it is a probability value, there is some probability that it is incorrect. If, for chance, a contig just happens to have a high cross correlation at some point where it is not supposed to go, it will ruin the entire matching process since it blocks the location of another contig which in turn will end up in the incorrect spot, blocking a third contig, and so on. The two other methods, found in section 7.3 and 7.4, mitigate this problem.

7.3 Tree Method

The main issue with the simplified approach is that a contig, that is matched at a bad spot by chance, can ruin the entire matching process. The method that is named the Tree method works around that. The assumption that the contigs cannot overlap is still kept and could potentially still be a problem.

The method matches contigs in different orders instead of just matching the contigs once. Each of these matching processes is called a branch, and all the branches make up a tree (hence the name of the method), which is the entire calculation process. Steps number 1-4 are the same as for the simplified approach, but after those the two methods are different. A visualization of the method can be found in Figure 13.

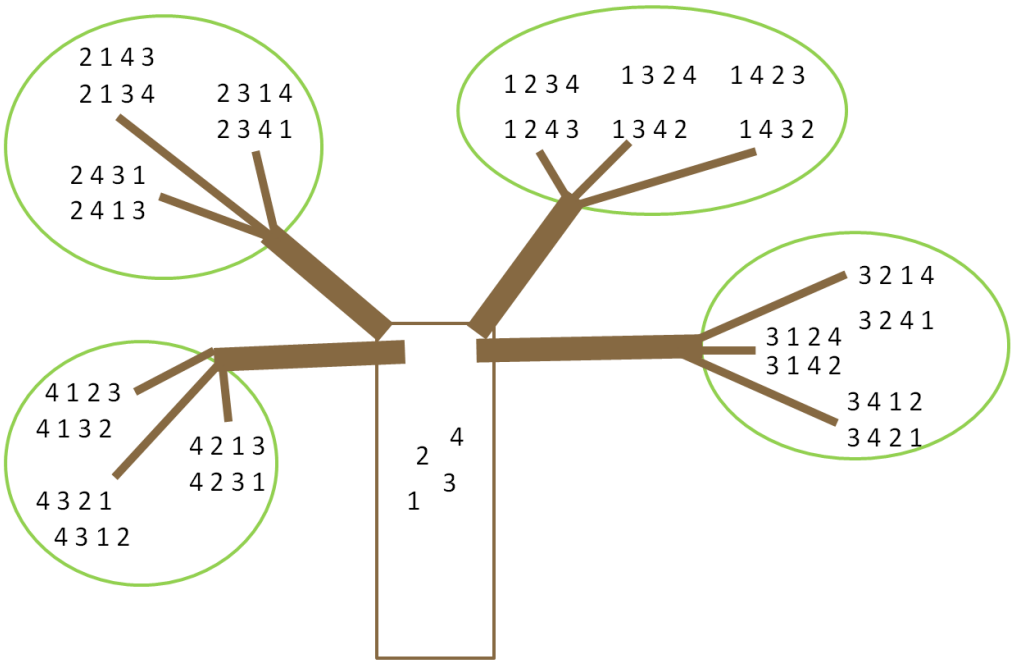


Figure 13. A visual representation of an example with four contigs of the tree method. At the start of the method there are four unmatched contigs (the middle of the tree), then the contigs are matched to the reference barcodes in a specific order. The result of one such matching is a branch. Some orders are closely related (like order 4, 1, 2, 3 and 4, 1, 3, 2) and are thus closer together in the tree.

Assume that there are N contigs that are supposed to be matched to an experimental barcode. The tree will then consist of $N!$ branches. The first branch will match contigs to the experimental barcode in order: 1, 2, 3, ..., N-1, N. The second will match contigs in order: 1, 2, 3, ..., N, N-1. This will continue for all permutations. If, for say, contig 4 happens to match incorrectly, all the branches that matches contig 4 will be ruined, since many of the other contigs will be placed incorrectly as well. The good thing with this method is that there are plenty of branches left. The best branches will be those when another contig has already taken up the spot where contig 4 would match incorrectly and thus making contig 4 having to move to another, hopefully correct, spot. An issue with this method is that all of a sudden there are plenty of branches and some of them are bad. How to sort out the good ones?

Each contig matched to the experimental barcode will have a p-value for the match. There is a method, called Fisher's method[12], that can be used in order to combine these to get a single p-value for the branch. First, the p-values are rescaled logarithmically and summed, using equation 7, where P_i is the p-value from equation 3 for contig *i*.

$$F = -2\sum_{i=1}^{N} \ln P_i \tag{7}$$

The sum, F, is χ^2 – distributed, with $2N$ degrees of freedom. By integrating from F to infinity over the distribution, similar to how p-values are calculated in section 5.2, a new p-value can be obtained. The distribution can be found in equation 8, Γ is the Gamma function.

$$\chi^2_{2N}(F) = \frac{1}{2^N \cdot \Gamma(N)} \cdot F^{N-1} \cdot e^{-\frac{F}{2}} \tag{8}$$

When every branch has a p-value, the branch with the lowest overall p-value should be the correct placement of the contigs. Another feature with this method is that two branches might be identical. If the tree method is used with perfect contigs, it would not matter the order which the contigs are matched since they would be matched to the correct spot. This is true since the contigs would not overlap and thus the order would not matter. Since the contigs rarely are perfect, due to edges being bad or experimental errors, a couple contigs will have several places where they match well. This will ensure that there are several branches each time. The most common branch, however, should also be the correct branch, since every time a contig is matched to the incorrect place, all the others have to match to incorrect places, but every time the contigs are matched to the correct place, the rest are also able to go to their correct places. Because there are many incorrect places, but only one correct, the number of identical, correct, branches will always increase, while an incorrect branch should not be identical with any other incorrect branches. Of course, two incorrect branches could be identical, but there is no reason for them to be identical and thus it would happen at random, while every correct branch must be identical since there is only one possibility.

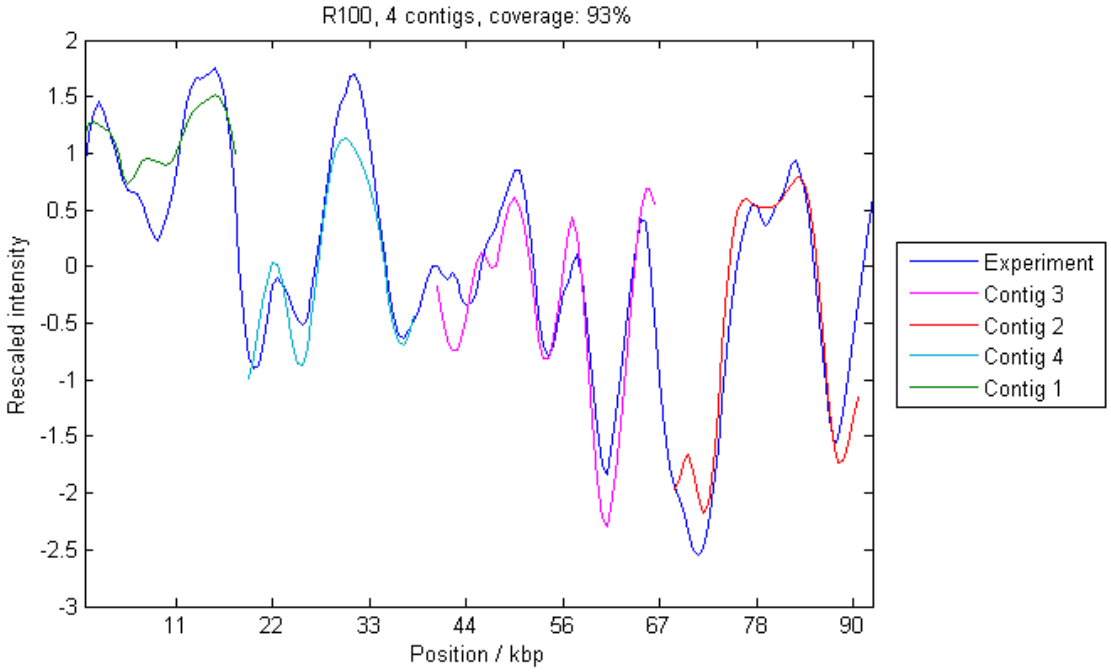


Figure 14. Four contigs from R100 matched using the tree method against an experimental barcode of R100. The contigs all seem to fit where they have been matched. By attaching the underlying sequences from the contig barcodes, the sequence for the entire R100 can be found.

An example of the results that can be obtained is found in Figure 14. What is seen there is an experimental barcode of R100 used as the reference for four contigs of R100. The contigs are of size 15 kbp all the way up to 25 kbp. The contigs were created by taking the well known sequence of R100 and chopping it into four pieces, thus ensuring that the order of the contigs was known. This means that contig 1 is supposed to be between contig 2 and 4, contig 2 is supposed to be between contig 1 and 3, and so on. Looking at the branch one can immediately see that the software managed to place them correctly on the barcode. Worth noting is that this was the best branch from the matching process, there were other suggestions which had only a couple correct, but this was the one with the lowest overall p-value.

The tree method is a robust way of placing contigs on an experimental barcode. The drawbacks are the computational time which is a factor $N!$ greater than that of the simplified approach, and that several possible solutions are found and it could be difficult to determine which one is the best. The assumption that contigs cannot overlap must also still be true.

7.4 Free Energy Method

Another method, that should only generate one, good, solution for the matching of many contigs to one experimental barcode, is called the Free Energy method. Free energy changes with both energy and entropy, and is a fundamental concept in statistical mechanics. Chemical reactions go to the state with lowest free energy, even in cases where that means losing energy (cooling down) in order to gain entropy. This loosely relates to the cost function used in this method.

Using equation 7, p-values can be converted into F-values, and the sum, as shown in equation 9, together with a cost for overlapping contigs, form a cost function E.

$$E = \sum_{i=1}^{N} F_i + \sum E_{overlap} \tag{9}$$

If during contig matching overlap is allowed, there will be a cost for each pixel that overlaps. This cost is used to discourage solutions with overlap since the contigs should not overlap if everything is done properly. By integrating over the distribution found in equation 8 using the cost E for a solution, a p-value for that solution is found. Repeating this procedure for all possible positions of contigs will give the best placement of the contigs. A visual representation of the F-value part of the Free Energy method can be found in Figure 15.

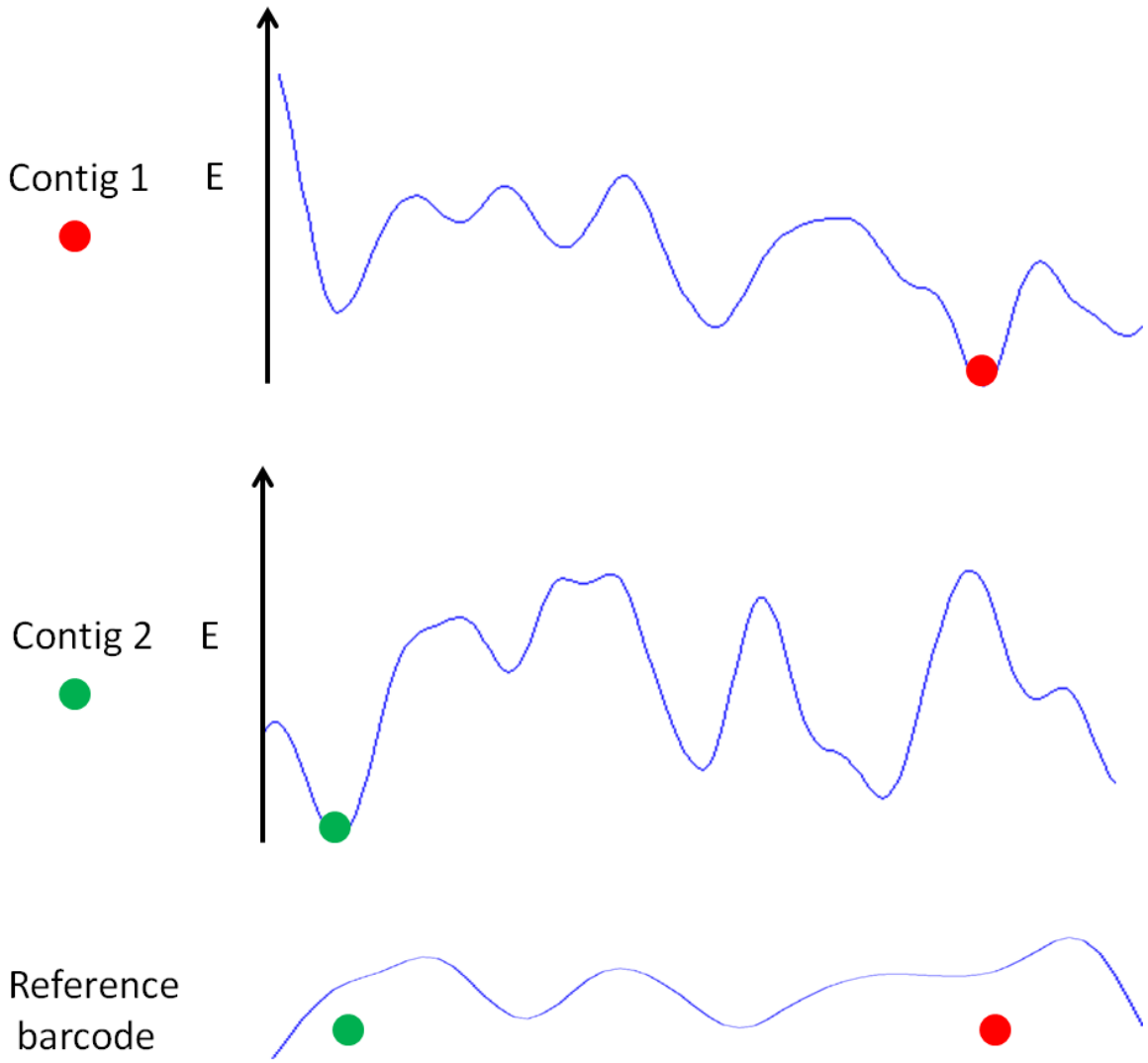


Figure 15. Visual representation of the F-value part of the Free Energy method. A cost landscape, using $\boldsymbol{F = -2 \ln p}$, where p is the p-value at a given position, is produced for each contig. By trying to minimize the sum of energies while also minimizing overlap, the optimal placement of the contigs can be found. In the figure above, both contig 1 and contig 2 are placed in the minimum cost location and do not overlap. This case would not require analysis to determine the optimal placement (which is shown in the row "reference barcode").

There is one big problem with this method, and that is computational cost. Assume that *N* contigs for a plasmid are going to be matched. An average plasmid can be up to 200 kbp or 350 pixels long. Since, in a "brute force approach", each start position (depending on plasmid length) must be examined, as well as the contig not being included at all, it requires 351^N calculations, where each calculation requires summation, both to figure out p-values for individual contigs and for the total p-value, and some tracking of positions. Then all of this data have to be saved as well. It is not impossible to have ten contigs for a 200 kbp plasmid. That means there will be $\sim 3 \cdot 10^{25}$ calculations. Even if each of these calculations was just adding a one to a variable, that would still take several hundred thousand millennia for one modern processor to finish calculating.

A way of mitigating the computational cost is a must if this method is to be used. One way of doing this is by calculating the p-value for each contig at every location on the experimental barcode first. After the calculations, a threshold is determined. All p-values higher than the threshold are

eliminated and only the ones below are considered. If a good threshold is used, the number of possible placements for each contig is reduced by a factor 16. Then the calculations are reduced to $21^{10} \approx 1.6 \cdot 10^{13}$, which only would take roughly 12 hours to finish. There is also a trade-off here regarding the p-value threshold. On one hand, a high threshold allows a higher number of possible sites for the contigs which increases the probability for a good match to be found, but, on the other hand, a lower threshold will reduce the computational cost to something reasonable. A threshold which only allows sites with p-value less than 0.2 has been used with good results, see Figure 18.

For a small number of contigs, it is possible to examine the entire cost landscape along the contigs position axis. In Figure 16, a heat map illustrating the cost for the special case when only two contigs were used with the free energy method. The cost is high for similar start positions for both contigs (i.e. the main diagonal), because they will have a large overlap and that increases the overlap term in the cost function. The overlap cost is 0.5 per pixel in this case. There are a couple of blue lines, both vertically and horizontally, and they show where one of the two contigs fits very well. For example, the blue vertical line at 122 on the x-axis indicates that contig 1 fits really well at that position, but (not as blue) similar lines can be seen at 37, 63 and 72. Similar can be seen for contig 2, for example at 68 on the y-axis.

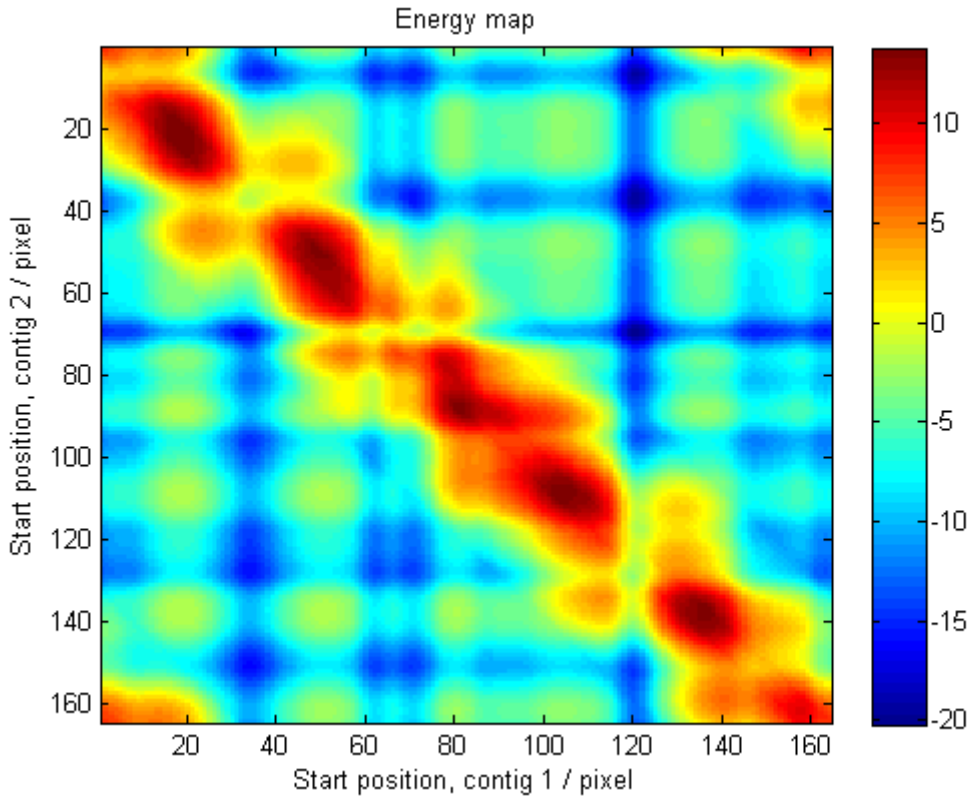


Figure 16. A two dimensional heat map showing the cost associated with placing (two) contigs on an experimental barcode, with overlap cost set to 0.5 per pixel. The optimal positions (or lowest cost) are at (122, 68). The contigs (and the experimental barcode) are from the plasmid R100.

From Figure 16, a couple of good solutions for the contig assembly puzzle can be found. With this method the two contigs could have overlapped, but the best solution (lowest cost) was at a position where they did not overlap because of the cost associated with overlap. To illustrate the effect of the overlap cost, another cost landscape for the same two contigs can be seen in Figure 17, but this time

with cost equal to zero for overlap. The minimum cost in the landscape is still at the same position (122, 68), but there are plenty of options to choose from (that was not viable in Figure 16). The cost for overlap can be tweaked a bit in order to not make overlap completely impossible, but still excluding most overlaps. If the experimentalists are confident that the experimental barcode and the contigs are perfect, the cost for overlap could be set to something high in order to exclude all overlap solutions.

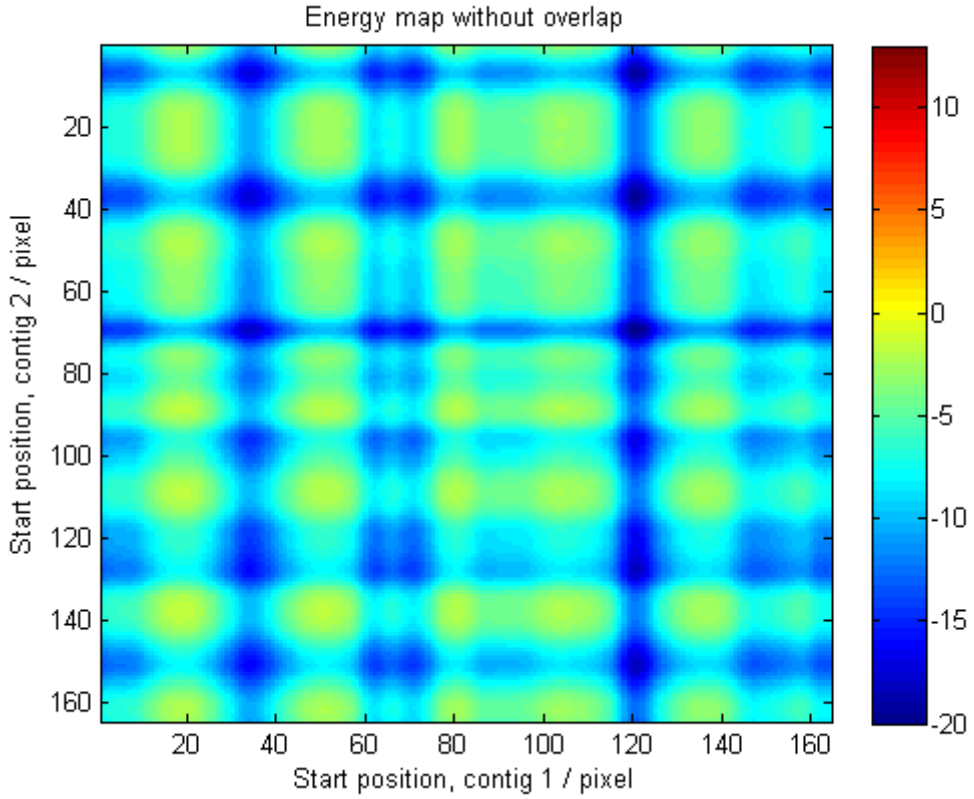


Figure 17. A two dimensional heat map showing the cost associated with placing (two) contigs on an experimental barcode, with overlap cost set to zero. The optimal positions (or lowest cost) are at (122, 68). The contigs (and the experimental barcode) are from the plasmid R100. Since overlap was not discouraged in this run, there are many positions that are almost equally likely for the contigs to occupy. The definition of a contig is a piece of sequenced DNA that has no overlap with the other contigs, and thus there are more false positives in this attempt.

In Figure 18, the same four R100 contigs as used in Figure 14 can be seen matched against an experimental R100 barcode, but this time the Free Energy method was used. Both cases are really similar, which suggests that the contigs should actually be placed in those positions. Since the sequence for R100 is known, it can actually be confirmed that the contigs are placed at the correct locations.

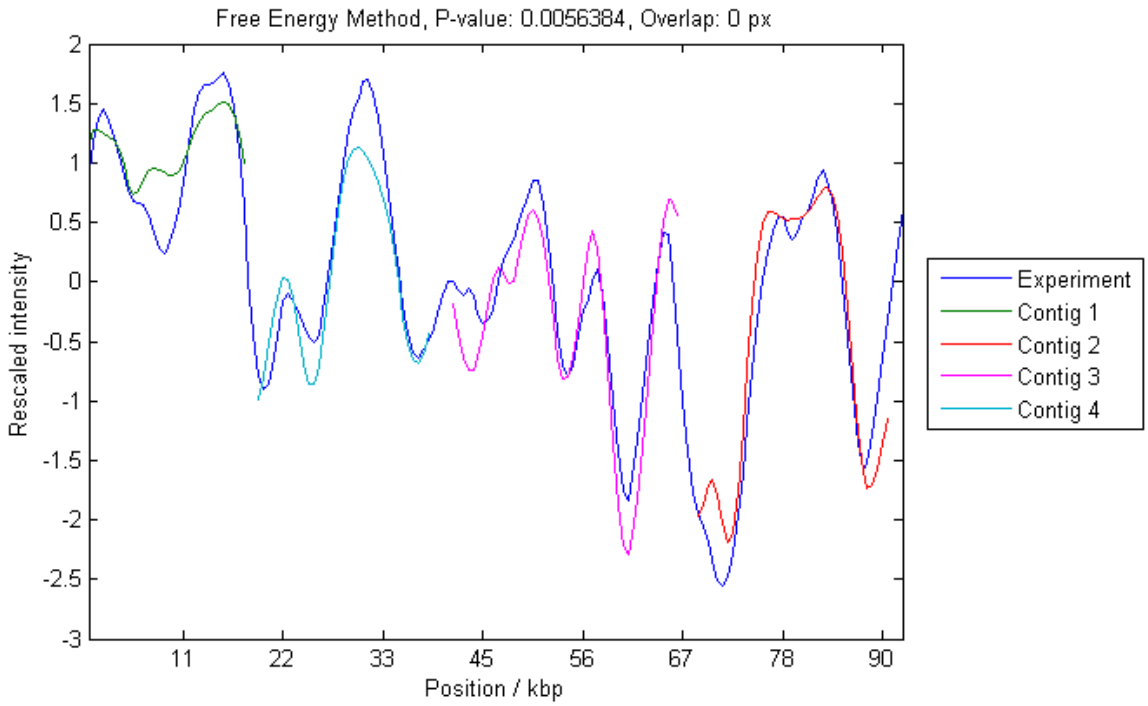


Figure 18. R100 contig barcodes matched against an experimental barcode from R100 using the Free Energy method. The overall p-value is less than 0.01 which indicates that the contigs are placed at good positions. By visually inspecting the match, it seems to be very likely that the contigs have been placed correctly.

7.5 Contig Assembly Problems

There are two main problems regarding the current methods of Contig Assembly. The first problem being that the contigs need to have a certain length in order to be matched correctly (otherwise the contigs will not be “unique enough”). The second problem is that that more contigs to match, means more possible ways of arranging them, and thus the computational cost increases and it might be impossible to get the results this century.

An example of the first problem can be seen in Figure 19, where the contigs from Figure 18 has been cut in half and then matched again against the experimental barcode of R100. According to Figure 18, the correct order should be 2, 1, 8, 7, 6, 5, 4, 3, and not 5, 3, 2, 6, 8, 4, 1 (with contig 7 being not matched at all). Two of the contigs are still placed in the correct positions, but the rest are not. Contig 4, the shorter of the two correctly placed contigs, are 12 kbp long in its raw form (before edge cutting). This seems to be some kind of minimum length for a contig to be useful, but even longer contigs could be useless if they do not contain a unique feature as contig 4 does.

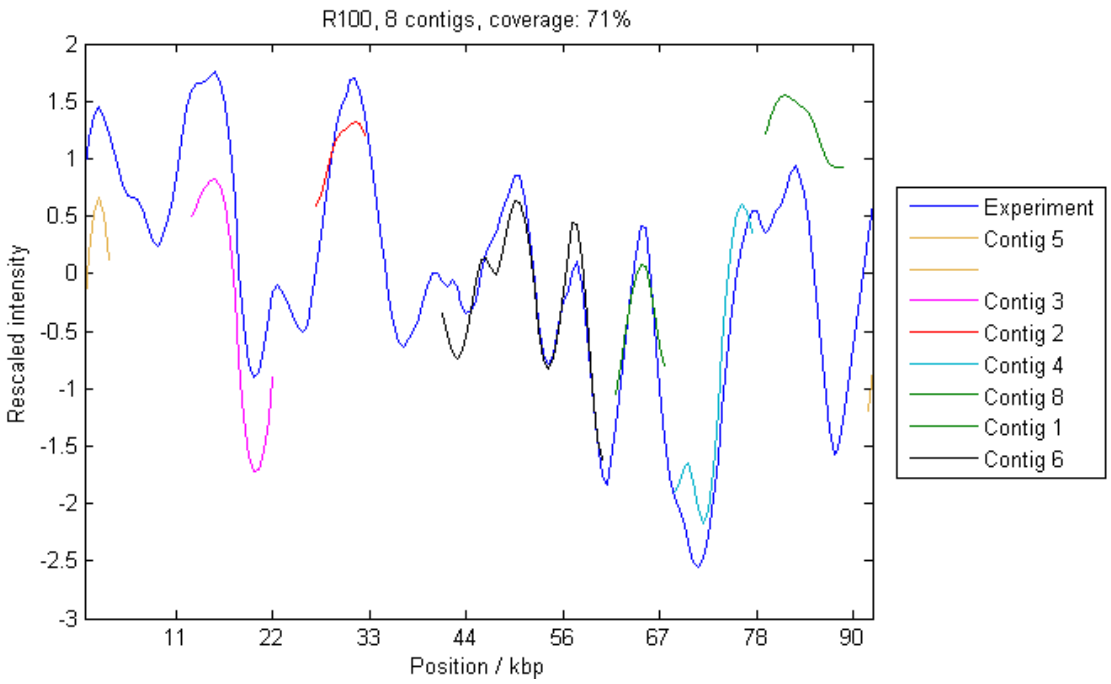


Figure 19. Best branch of an experimental R100 barcode matched against 8 R100 contigs. Some of the contigs are too short to be unique, and are thus matched at incorrect locations. From this match, the correct sequence for the whole molecule is not possible to extract. A possible way to remedy this, would be by randomly attach two contig sequences to each other and see if the new contig barcode would be get a good match. If not, the sequence should not be like that. This would be time consuming and not guaranteed to generate useful results. Best results are obtained from longer contigs.

The second big issue for the contig assembly is the computational cost. A comparison between the methods has been made in order to visualize the relative cost difference. Assume that testing one location for a contig has a computational cost of 1 arbitrary unit, the length of the experimental barcode is l_{exp} and there are N contigs. The cost for the Tree method would then be $c_{tree} = \left(l_{exp}\, N^2\right) N!$, since there are $N!$ branches and each branch matches all contigs at every location (which is $l_{exp}N$ times), chooses one and then matches the rest one more time and so on. The cost for the Free Energy method was already calculated in section 7.4, and was $\left(l_{exp}\right)^N$. By applying a low enough p-value threshold to reduce number of sites that the Free Energy method has to look through, the cost can be reduced by some factor to the power of N. A, rough, comparison between the cost of the methods, including costs if different p-value thresholds are used, can be seen in Figure 20. The cost axis is logarithmically scaled to better show all the trends. For a low number of contigs, the Free Energy method could actually be quicker than the Tree method. When the number of contigs increases, the Free Energy method's cost also increases quickly. Unless a p-value threshold that reduces the number of sites with a factor 20 is applied, the Tree method will be quicker for more than a few contigs.

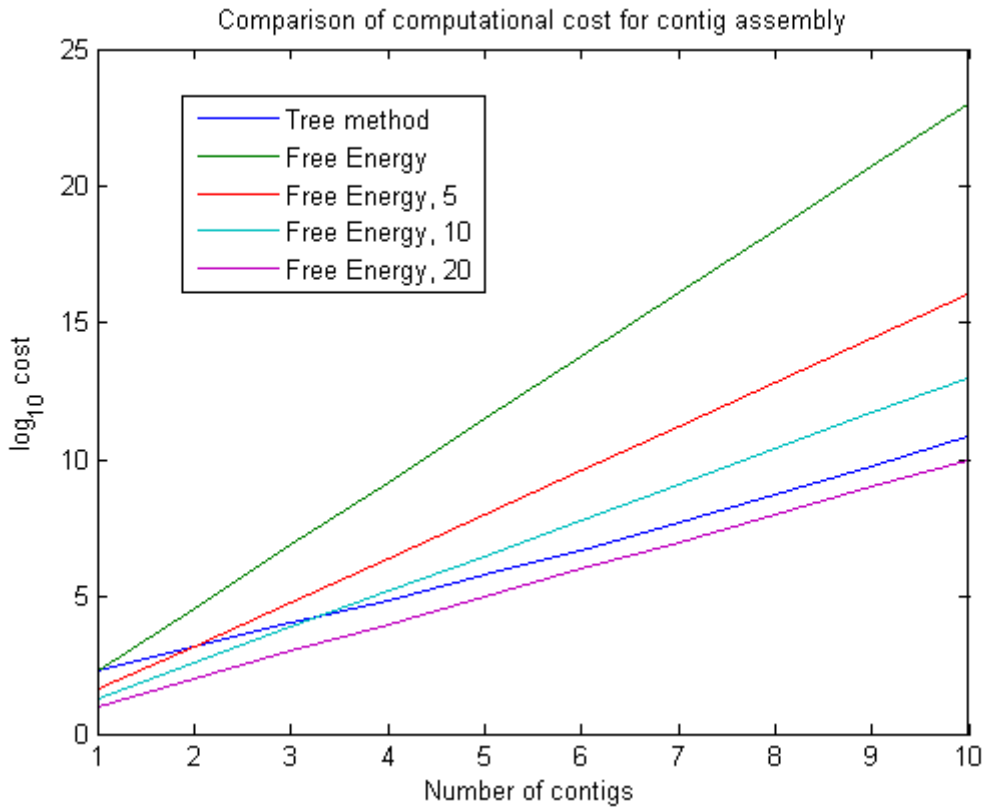


Figure 20. Computational cost for the contig assembly methods using an experimental barcode of length 200 pixels. The number denoted in the legend next to “Free Energy” represents the factor with which the number of possible locations is reduced by applying a p-value threshold. The cost axis is logarithmically scaled. The “Tree method” or the “Free Energy, 20” method should be used for number of contigs higher than four otherwise the computational cost is too high.

8. Conclusion and Outlook

The methods developed both for barcode comparison and for contig assembly shows positive results, and could be useful methods in future work. The barcode comparison method worked for the used set of data and probably will work for similar data. The plasmid containing the antibiotic resistance gene could be found in all samples from patients showing signs of being infected by an antibiotic resistant strain of bacteria.

The contig assembly on the other hand works in principle, but not practically. When assembling the contigs using different lengths, a lower limit on length of the contigs could be approximated to 12kbp, independent on method (Tree method or Free Energy method). Since the lower limit on the contig length is quite high, the use right now is limited.

8.1 Barcode comparison method conclusion

The results for the barcode comparison method used to track the outbreak at Sahlgrenska University Hospital, both when asking Q1 ("are these barcodes the same?") and Q2 ("are any barcode a part of another barcode?") seem to be working just fine. When Q1 is asked, as seen in Figure 8, the software distinguishes between barcodes and sorts them into groups. Even though the results might not be as clear for Q2 as for Q1, the results show potential by turning a poor match from Q1 into a decent match when double circularly permutations were allowed. Something worth noting regarding the p-values is that, since there will be more possibilities for matches, the cross-correlation should be higher when asking Q2 than Q1. This introduces the possibility of over-fitting since the information in the barcodes can be shifted around. The zero-model is thus calculated using random barcodes compared to the experiments with the same possibility of over-fitting. This shifts the extreme value distribution peak towards higher cross correlation values and counter-acts the drop in p-values because more possibilities are given.

A future step would be to develop a third method that would detect multiple regions of inserted genes (the current method identifies a maximum of one inserted region of genes). The main problem with detecting these regions is the computational cost. For each region that is going to be found, the computational cost increases rapidly. The problem resembles the Free Energy contig assembly problem. Instead of matching contigs to an experimental barcode, it is "holes" that are being matched to the barcode instead. These holes should be of various sizes, although the sum is constant (and equal to the difference in length between two barcodes). Assume there are N holes, with total length l, and k is the length of the longer barcode. The computational cost would not only scale as k^N, like the Free Energy method, but as $\binom{l+N-1}{N} k^N$ because all possibilities for sizes of the holes would have to be tried as well.

Another possible application for the barcode comparison method that has not been tried yet would be gene identification. If a barcode containing just one gene is obtained, it could be compared with other barcodes, e.g. from plasmids, in order to determine if that bacterium is carrying that specific gene. It could also work if a gene's sequence is known, but then by making a theoretical barcode from it before comparing. A concern here would be that the gene has to be long enough so that the information would not be too distorted by blurring effects and edge problems, making the barcode only having non-unique features, similar to what is seen in Figure 19.

8.2 Contig assembly methods conclusion

The branch, from the Tree method, that can be seen in Figure 14 shows good result for fairly long contigs, but many contigs can have a length of less than 10 kbp. These contigs are often too short to contain any useful information; an example can be seen in Figure 19.

Even though the methods are working, the experimentally created contigs must have sufficient lengths. The limiting factor, right now, is the width of the PSF. If there were no PSF to blur the intensity curve, shorter contigs could be matched. After the PSF it is probably the pixellation effect that reduces the information the most.

Looking at the result, the Tree method is enough for assembling longer contigs, and is much quicker than the Free Energy method. The good results are only found after looking through some branches and picking the best one and since the Free Energy method only finds one solution, it should be preferable to use that one. By setting a p-value threshold for the Free Energy method, it is possible to speed up that method as previously stated, but with the possibility to actually miss the global cost minimum by excluding a position that would not be the optimal position for one contig but would be the best when considering all of them (because of overlap costs). There is some possibility to do the same for the Tree method by not restarting the matching process for each branch, but instead reuse some of the previous matching; this solution would require more computer memory, but less processing power. This is not implemented yet, but could be a future project.

Another way of mitigating the computational time (without optimizing the computational cost), would be to parallelize the computing. Since each branch in the Tree-method and each combination of positions in the Free Energy-method are independent of each other, the calculations could be divided over multiple processors. This would speed up the computational time by a factor equal to the number of processors used (minus some time due to overhead cost).

Disregarding computational cost, the Free Energy method is probably the most accurate contig assembly method for two reasons. First being that it tests every possibility and presents the best one. It could also be made to present the ten best solutions and then the person using the method could use visual inspection to single out the best solution. The second reason is that there are no assumptions about that the contigs cannot overlap, meaning this method finds solutions that the other methods would not. The big drawback is its current high computational cost, which is mitigated by the use of a p-value threshold.

9. References

1. Ronaghi, M., M. Uhlén, and P. Nyrén, *A sequencing method based on real-time pyrophosphate.* Science, 1998. **281**(5375): p. 363-365.
2. Kawashima, E., L. Farinelli, and P. Mayer, *Labelled nucleotides incorporated stepwise in a nucleic acid molecule by primer extension*. 1998, Google Patents.
3. Parab, H.J., et al., *A gold nanorod-based optical DNA biosensor for the diagnosis of pathogens.* Biosensors and Bioelectronics, 2010. **26**(2): p. 667-673.
4. Reisner, W.W., H. Flyvbjerg, and J.O. Tegenfeldt. *Melting mapping in nanochannels*. in *International Conference on Miniaturized Systems for Chemistry and Life Sciences, San Diego, CA*. 2008.
5. Nilsson, A.N., et al., *Competitive binding-based optical DNA mapping for fast identification of bacteria-multi-ligand transfer matrix theory and experimental applications on Escherichia coli.* Nucleic acids research, 2014: p. gku556.
6. Nyberg, L.K., et al., *A single-step competitive binding assay for mapping of single DNA molecules.* Biochemical and biophysical research communications, 2012. **417**(1): p. 404-408.
7. Noble, C., et al., *A fast and scalable algorithm for alignment of optical DNA mappings.* arXiv preprint arXiv:1311.6379, 2013.
8. Lena K. Nyberg, S.Q., Gustav Emilsson, Nahid Karami, Erik Lagerstedt, Vilhelm Müller, Charleston Noble, Susanna Hammarberg, Adam N. Nilsson, Fei Sjöberg, Joachim Fritzsche, Erik Kristiansson, Linus Sandegren, Tobias Ambjörnsson, Fredrik Westerlund, *Rapid identification of intact bacterial resistance plasmids via optical mapping of single DNA molecules.* Submitted.
9. Pinheiro, E.C. and S.L. Ferrari, *A comparative review of generalizations of the Gumbel extreme value distribution with an application to wind speed data.* Journal of Statistical Computation and Simulation, 2015: p. 1-21.
10. Schreiber, T. and A. Schmitz, *Surrogate time series.* Physica D: Nonlinear Phenomena, 2000. **142**(3): p. 346-382.
11. Huang, X., *A contig assembly program based on sensitive detection of fragment overlaps.* Genomics, 1992. **14**(1): p. 18-25.
12. Fisher, R.A., *Statistical methods for research workers*. 1925: Genesis Publishing Pvt Ltd.

1. Appendix A – Zero Model detailed method

In the main text, in section 5, a way of quantifying barcode matches was discussed. In this appendix, more explicit details of this method will be presented.

When generating a Zero Model, the goal is to find the PDF of cross correlation values when matching two barcodes assuming the two barcodes has nothing to do with each other. This is accomplished by generating two Zero Models for each pair of barcodes, one when comparing the first barcode to phase randomized versions of the second one, and another when reversing the roles. A barcode (with index i) is characterized through an intensity vector, $I_i(x)$. This vector has n_i elements. The Zero Model for barcode i and barcode j is called ZM_{ij} and is not the same as ZM_{ji}, as previously stated. The input parameters to generate ZM_{ij} is $I_i(x)$, n_j, $I_{thy}(f)$, k_i, k_{thy} and a stretch factor (and number of stretches). $I_{thy}(f)$ is the Fourier Transformed (FT) barcode of all the theory barcodes (3224) found in a theory database. The k_i and k_{thy} is the kbp/pixel values for barcode i and for the general FT barcode respectively. Note that for Q1 all n_j are independent on j (since all barcodes are stretched to the same length), and thus all ZM_{ij} will be independent on j.

The steps of the method are found in the list below. After the list, each step is discussed.

1. If $I_i(x)$ is supposed to be stretched, all the stretched versions are generated.
2. $I_{thy}(f)$ is stretched (interpolated) to length a.
3. The negative frequencies are "folded" over the positive frequencies (mirrored and replaced).
4. Random phase factors are symmetrically (around f=0) multiplied to $I_{thy}(f)$.
5. The new $I_{thy}(f)$ is inverse Fourier Transformed to form a PR barcode, $I_{PR}(x)$.
6. $I_{PR}(x)$ is stretched to length n_j.
7. $I_i(x)$ and $I_{PR}(x)$ are compared and the best cross correlation value is found and saved.
8. If $I_i(x)$ was stretched, all the versions are compared to $I_{PR}(x)$ and the best of the best cross correlations are used instead of the best for the non-stretched version.
9. Repeat steps 4-7, 1000 times.
10. The mean and variance of all the best cross correlation values are found.
11. Using the mean and variance, μ and β for a Gumbel distribution is determined.

The first step is only used for Q2, otherwise all barcodes are the same length, and no stretching is required. The length n_{thy}, are usually not the correct length, n_j, that $I_i(x)$ is supposed to be compared against, and thus it has to be changed. Another factor to consider is that $k_i \neq k_{thy}$ in the general case, and thus k_{thy} has to be changed as well. By interpolating on a barcode (not on the Fourier Transform), k may be changed. Call the factor of length change in the $I_{PR}(x)$, b. Since $b = \frac{k_{thy}}{k_i}$, the length that $I_{PR}(x)$ needs to have can be calculated $a = \frac{n_j}{b}$. Since $a \neq n_{thy}$ in a general case, $I_{thy}(f)$ needs to be stretched to length a, before it is transformed into a barcode (otherwise the k_{thy} will be changed twice and it will not assume the correct value).

The third step is a way of removing interpolation differences. The interpolation will cause $I_{thy}(f) \neq I_{thy}(-f)$, even though it is only by a very small difference. Since the Fourier Transformed barcode needs to be symmetrical in order to generate a real barcode, this trick is done. The fourth step uses similar reasoning. Random phase factors are generated for the negative frequencies and then the same phase factors are then multiplied symmetrically to the positive frequencies.

In step six, $I_{PR}(x)$ is stretched to length n_j, since $I_{PR}(x)$ is supposed to represent a random version of $I_j(x)$. The following two steps save the best cross correlation values in order to find a Gumbel distribution, which is the Zero Model. The μ and β found in step 11 then defines the CDF, used to determine the p-value for a match between $I_i(x)$ and $I_j(x)$.

2. Appendix B – Alternative to Gumbel Zero Model

If all the cross correlation values obtained from a comparison between two barcodes are considered to be random numbers with a Gaussian distribution, predicting the extreme value can be done using equation (7)[5], where $z = \frac{C-\mu}{\sigma\sqrt{2}}$, k is the order of value (k=1 is the highest value) and N_{eff} is the effective sample size (or otherwise known as the number of cross correlation values, for each comparison, that is uncorrelated). Two examples of the extreme value distribution can be found in Figure 21 and Figure 22, where it is shown how the N_{eff} parameter affects the modeled distribution.

$$\Phi(C') = e^{-z^2}\left(\frac{1}{2}(1-\operatorname{erf} z)\right)^{k-1}\left(\frac{1}{2}(1+\operatorname{erf} z)\right)^{N_{eff}-k} \quad (7)$$

All that is needed to calculate the distribution are μ, σ and N_{eff}. The last one, N_{eff}, is a bit tricky to determine. One way to determine N_{eff} is to compare many "random" barcodes (discussed more in section 5.3), using at least a 1000 to get good statistics, to barcode 1 and then saving all the cross correlation values in one histogram and all the extreme values in another. By finding the mean and the standard deviation of the histogram containing all the cross correlation values, assumed to be Gaussian distributed, μ and σ is obtained. By then fitting equation (7) to the extreme value histogram, using the least square method, N_{eff} is determined. These three parameters can then be used to calculate the Zero Model which then can be used in equation (3), together with a cross correlation value from the comparison of barcode 1 and 2, in order to calculate a p-value.

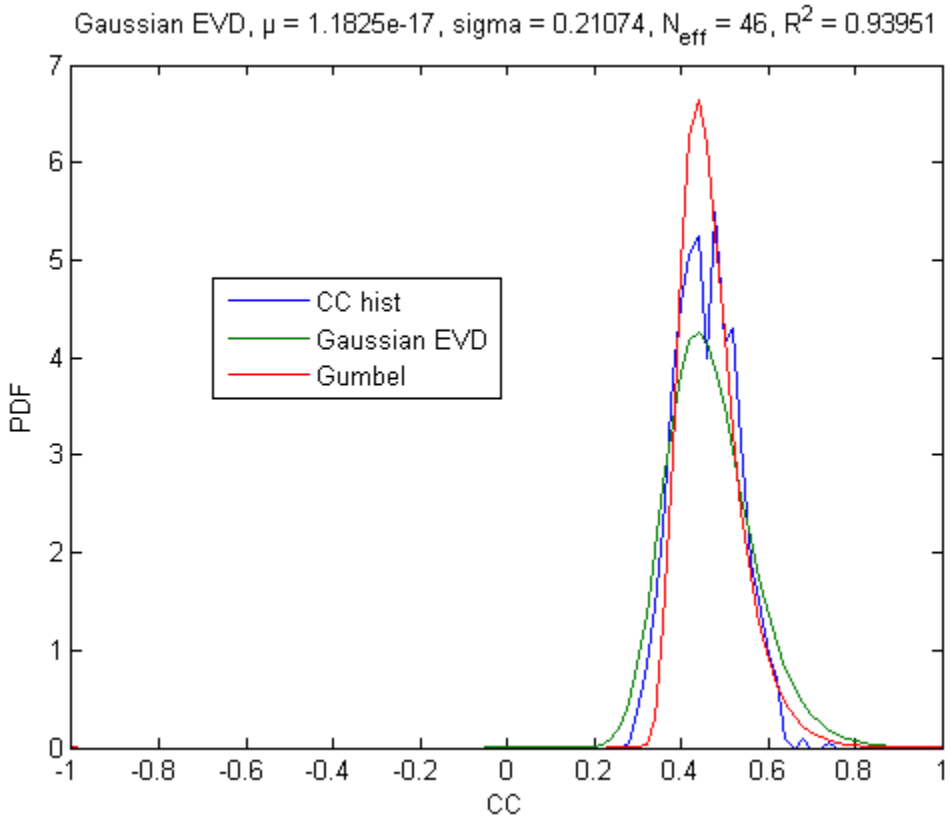


Figure 21. (Blue line) Histogram of cross correlations between 1000 Phase Randomized barcodes and an experimental barcode (S_P1a). (Green line) Extreme Value Distribution for Gaussian distributed values, given a mean (μ), variance ($\boldsymbol{\sigma^2}$) and effective sample size ($\boldsymbol{N_{eff}}$) (taken from the cross correlation distribution). A good

it ($\boldsymbol{R^2}$ > 0.9), and the best one found when testing different values of $\boldsymbol{N_{eff}}$. (Red line) Gumbel distribution, created by extracting a $\boldsymbol{\beta}$ and $\boldsymbol{\kappa}$ from the cross correlation distribution. $\boldsymbol{R^2}$ is the coefficient of determination.

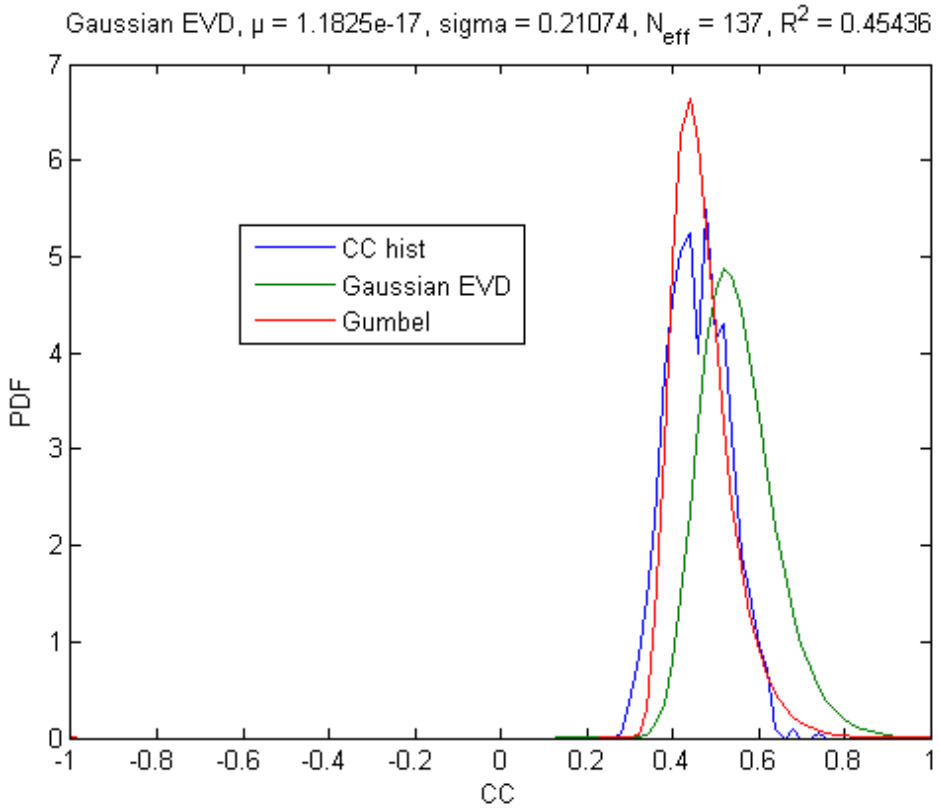


Figure 22. (Blue line) Histogram of cross correlations between 1000 Phase Randomized barcodes and an experimental barcode (S_P1a). (Green line) Extreme Value Distribution for Gaussian distributed values, given a mean (μ), variance ($\boldsymbol{\sigma^2}$) and effective sample size ($\boldsymbol{N_{eff}}$) (taken from the cross correlation distribution). Not the optimal fit ($\boldsymbol{R^2}$ < 0.9), just an illustration of the effect of changing $\boldsymbol{N_{eff}}$. Optimal fit is found in Figure 21. (Red line) Gumbel distribution, created by extracting a $\boldsymbol{\beta}$ and $\boldsymbol{\kappa}$ from the cross correlation distribution. $\boldsymbol{R^2}$ is the coefficient of determination.

The reason this method was discarded was that it is not certain that the cross correlation values are distributed as a Gaussian. Several tests shows that the distribution is very similar and, as can be seen in Figure 21, the fit to the histogram of cross correlation values is quite good, but the fit to a Gumbel distribution is better. One important difference might be that there is no need to estimate N_{eff} in the Gumbel case and thus it is much easier to get the correct distribution for that one.

Something else that also was tested was using equation 7 with $k \neq 1$, and looking at distributions of the second best cross correlation value, or the third (and so on), in order to more accurately determine N_{eff}. The method proved worse than the one described in this appendix, since the distributions for all but the extreme value were even less sensitive to N_{eff} and with a limited number of random cross correlation values, something with high sensitivity would be required.

3. Appendix C – Graphical User Interface development

Alongside developing the methods used in the thesis, a software with a Graphical User Interface (GUI) has been created using Matlab. The software is used regularly by the experimentalists working in Westerlund Lab at Chalmers. Each method has its own software, and section 3.1 will cover the software which compares barcodes to each other (used in section 5), while section 3.2 will talk about the software used for contig assembly in section 7.

3.1 GUI for Experimental barcode comparison

When starting the software, the user is immediately prompted to select which barcodes to compare. If only one (or zero) barcodes are chosen, the software sends an error message telling the user to select more barcodes before it shuts down itself. The second step, shown in Figure 23, allows the user to input how the lengths of the barcodes are to be calculated. The first option (from the left) stretches all the barcodes to the same length, and implicitly asks Q1 (“are these barcodes the same?”), since the barcodes compared are of the same size. The second and third option stretches the barcodes to individual inputted lengths. Since there are experimental deviations in the kilo base pair (kbp) per pixel value between experiments and not all plasmid lengths are known, the second option allows the user to use the calculated kbp/pixel value obtained from a known reference molecule instead of actually inputting the length of each experimental barcode. Both the second and the third option implicitly ask Q2 (“are any barcode a part of another barcode?”), since the barcodes could be stretched to different lengths and thus they cannot be the exact same.

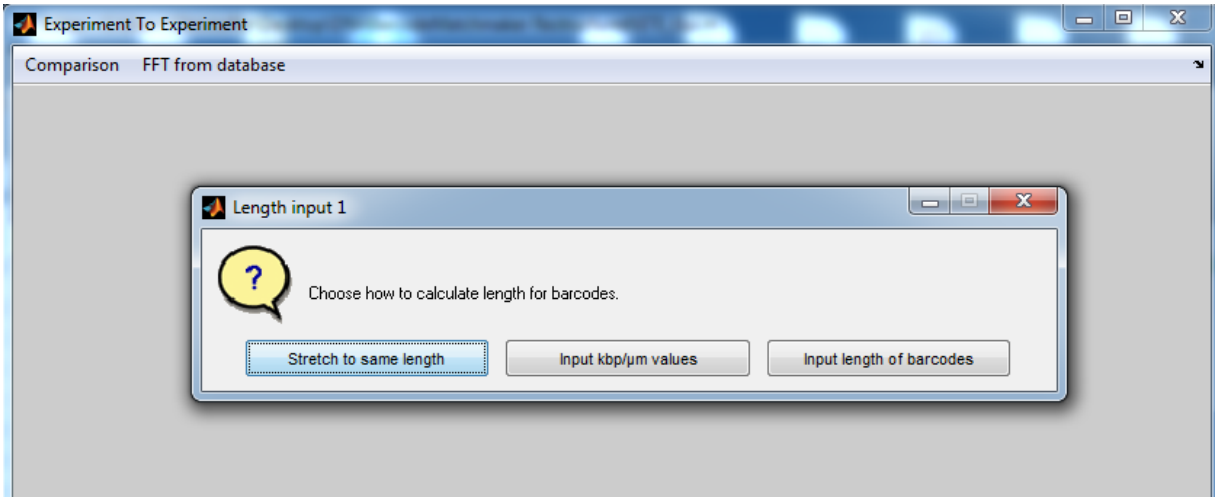


Figure 23. Length option input dialog for the Experiment To Experiment software. The first option (counting from the left) leads to Q1 (“are these barcodes the same?”) comparison, while option two and three lead to Q2 (“are any barcode a part of another barcode?”).

If the first option is chosen, the software asks for a length to assign to all the barcodes (in order to calculate the kbp/pixel for them). The other two options lead to another dialog window asking for the individual values of either kbp/pixel or kbp for each barcode. After this step, the software wants to know if the barcodes are circular, which is true for plasmid barcodes. This affects how the barcodes are compared to each other (i.e., if pixel 1 for barcode 1 can be matched anywhere to barcode 2 or if it only can match so that the two barcodes fully overlap).

When the user has entered all the input for the experimental barcodes the software moves on to the next set of input options. These inputs are for the Zero Model and can be seen in Figure 24. There are

three options to choose from depending on which type of Zero Model (description of Zero Model is found in section 5.2) the user wants to compare the barcodes against. The first option (from the left) generates 1000 random sequences (with the same length as the longest experimental barcode that was chosen previously) and converts them into theoretical barcodes and these barcodes are the "random" barcodes used to generate the Zero Model. The second option allows the user to enter barcodes which are then Fourier Transformed and then phase randomized in order to generate the barcodes for the Zero Model (see section 5.3 for details). The third option allows the user to enter an already processed Fourier Transform of Zero Model barcodes (that can be generated from the second drop down menu in the upper left corner in Figure 24).

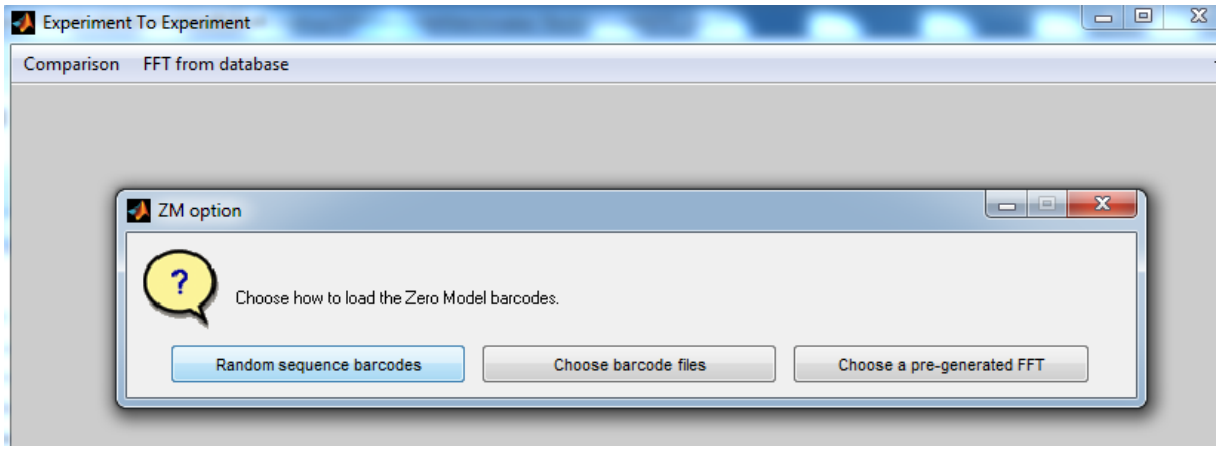


Figure 24. Zero Model option input for the Experiment To Experiment software. The first option (from the left) generates theoretical barcodes from random sequences, while the other two options use barcodes chosen by the user, in order to generate the Zero Model.

When all the options regarding the Zero Model have been inputted, the software starts comparing the experimental barcodes (as described in section 5 and 6). When the calculations are completed, a result window is shown, similar to the example that is found in Figure 25. The matrix shows the P-value for a match for each barcode compared to the others (including compared to itself). There are two buttons "Plot barcodes" and "Save barcodes txt", that takes the indices from the two boxes in the lower left corner and either plots the barcodes (example can be seen in Figure 26), or saves the intensity values in each pixel, when they are aligned as good as possible, to a txt-file. The last three buttons saves the p-value matrix, the underlying (not displayed) cross correlation matrix or a mean of p-value from matching barcode i with j and j with i.

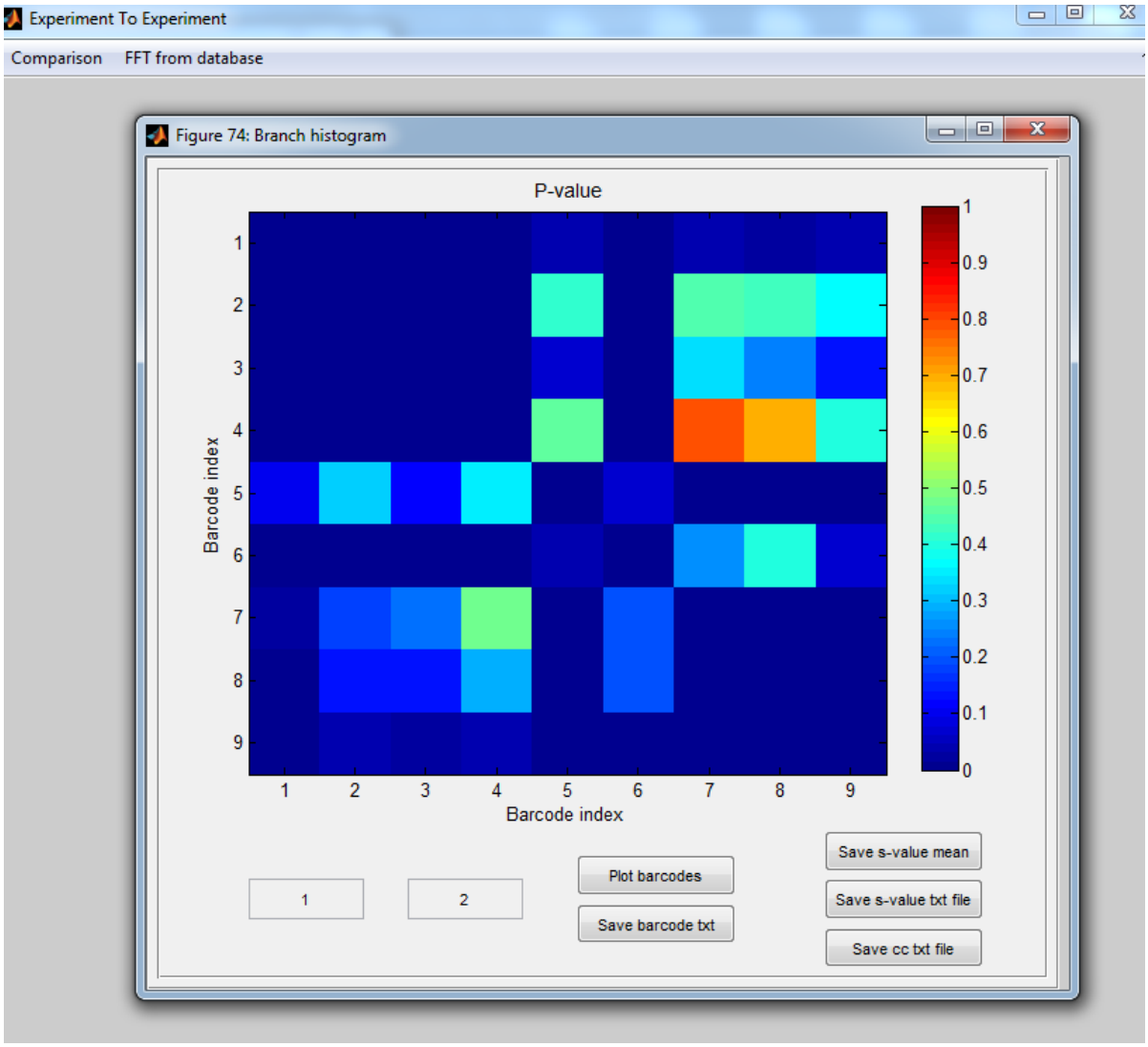


Figure 25. Result window for the Experiment To Experiment software. The matrix shows the p-value for a match between for each barcode chosen at the start of the software. All of the buttons saves information in txt-files, except for “Plot barcodes”, which plots the barcodes specified in the two text boxes in the lower left corner.

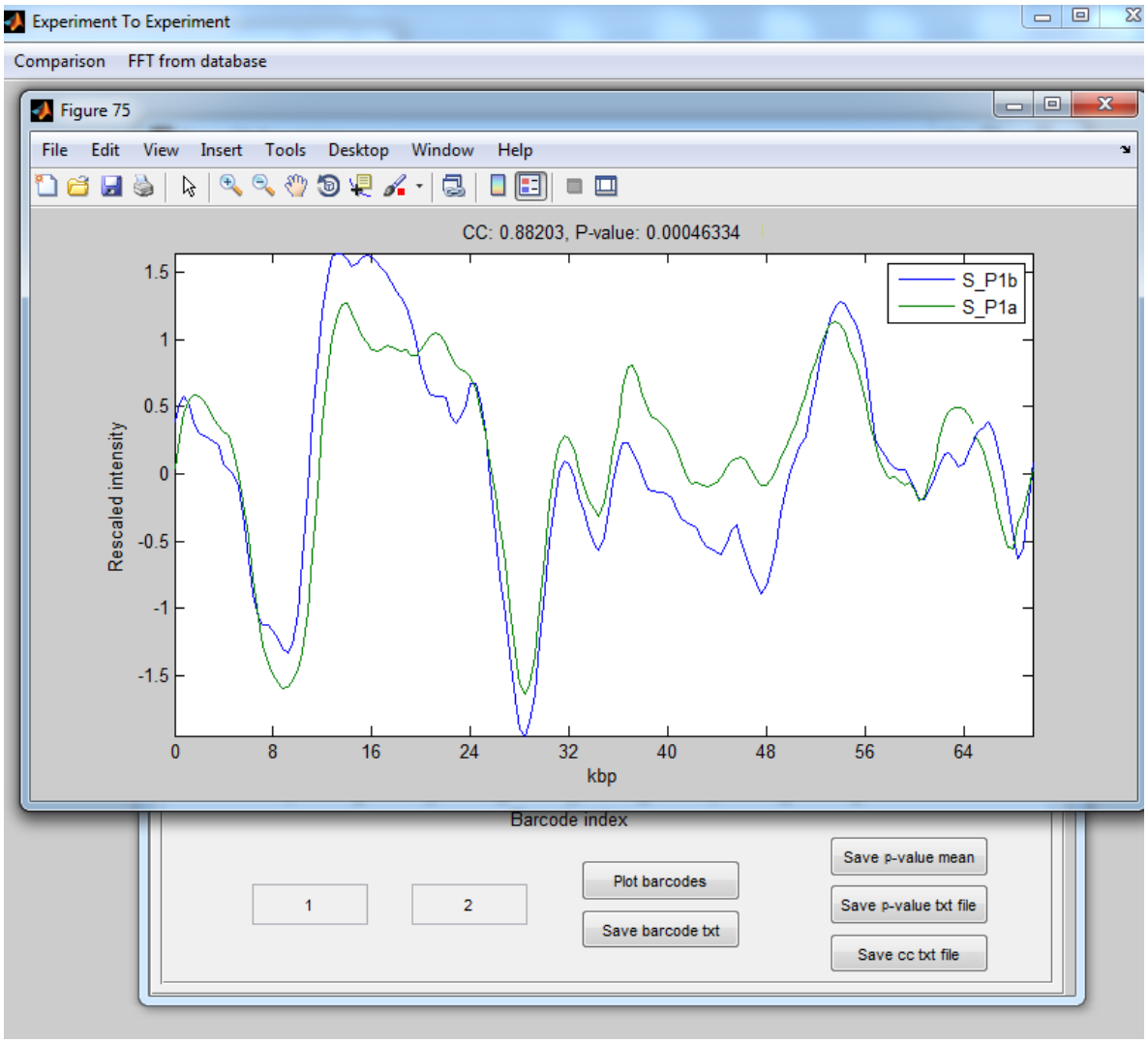


Figure 26. Two matched barcodes plotted against each other using the GUI of the Experiment To Experiment software. The image was generated by pressing the button labeled “Plot barcodes” in Figure 25. The names of the barcodes are also displayed in the upper right corner of the plot (unlike when the barcodes are chosen where they are referred to by their index).

3.2 GUI for Contig Assembly

There are two Contig Assembly methods developed in this thesis (Tree method and Free Energy method), and they both have a software associated with them. From the drop down menu that can be seen in the top left corner in Figure 27, the user can choose which method to use. This software was, similar to the Experiment To Experiment software from section 3.1 in Appendix C, created in Matlab. The Graphical User Interface (GUI) for the two softwares is quite similar and for the purposes of this demonstration, only the Free Energy method’s software is shown.

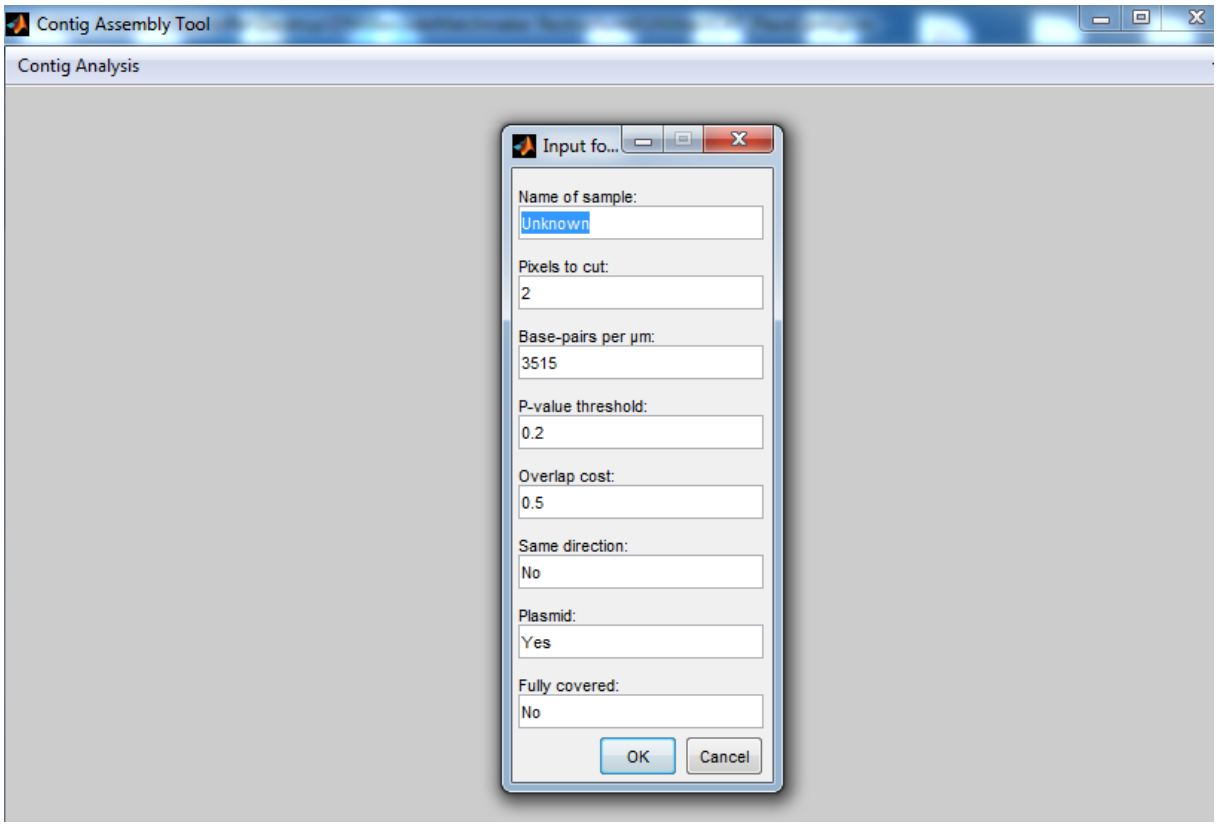


Figure 27. Configuration input for the Free Energy version of the Contig Assembly software. Both options about how the theoretical barcodes should be constructed as well as information about the matching process has to be chosen.

The software starts by opening two dialog windows where the user is supposed to choose the experimental barcode used as a template and the file(s) containing the contigs. After that, the menu found in Figure 27 can be seen. In the menu all things related to the contigs can be chosen, such as base pairs/µm and the number of pixels that will be cut on the edges (discussed in section 7.1), but also information if all the contigs are confirmed to have been read in the same direction or if the entire experimental barcode has been covered by the contigs.

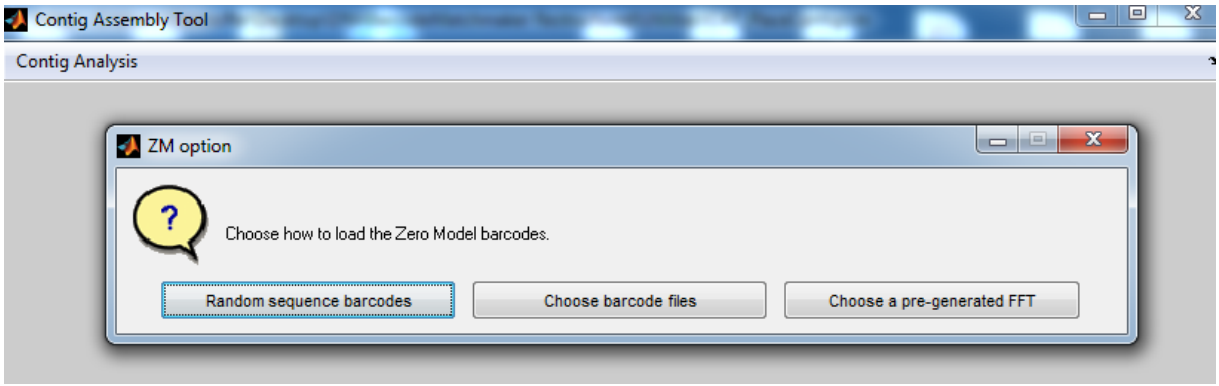


Figure 28. Zero Model input for the Contig Assembly software. Works in the same way as the ZM option found in Figure 24.

The next window allows the user to choose which Zero Model that is going to be used in order to calculate p-values for the matching process. This works exactly in the same way as described in section 3.1 in Appendix C.

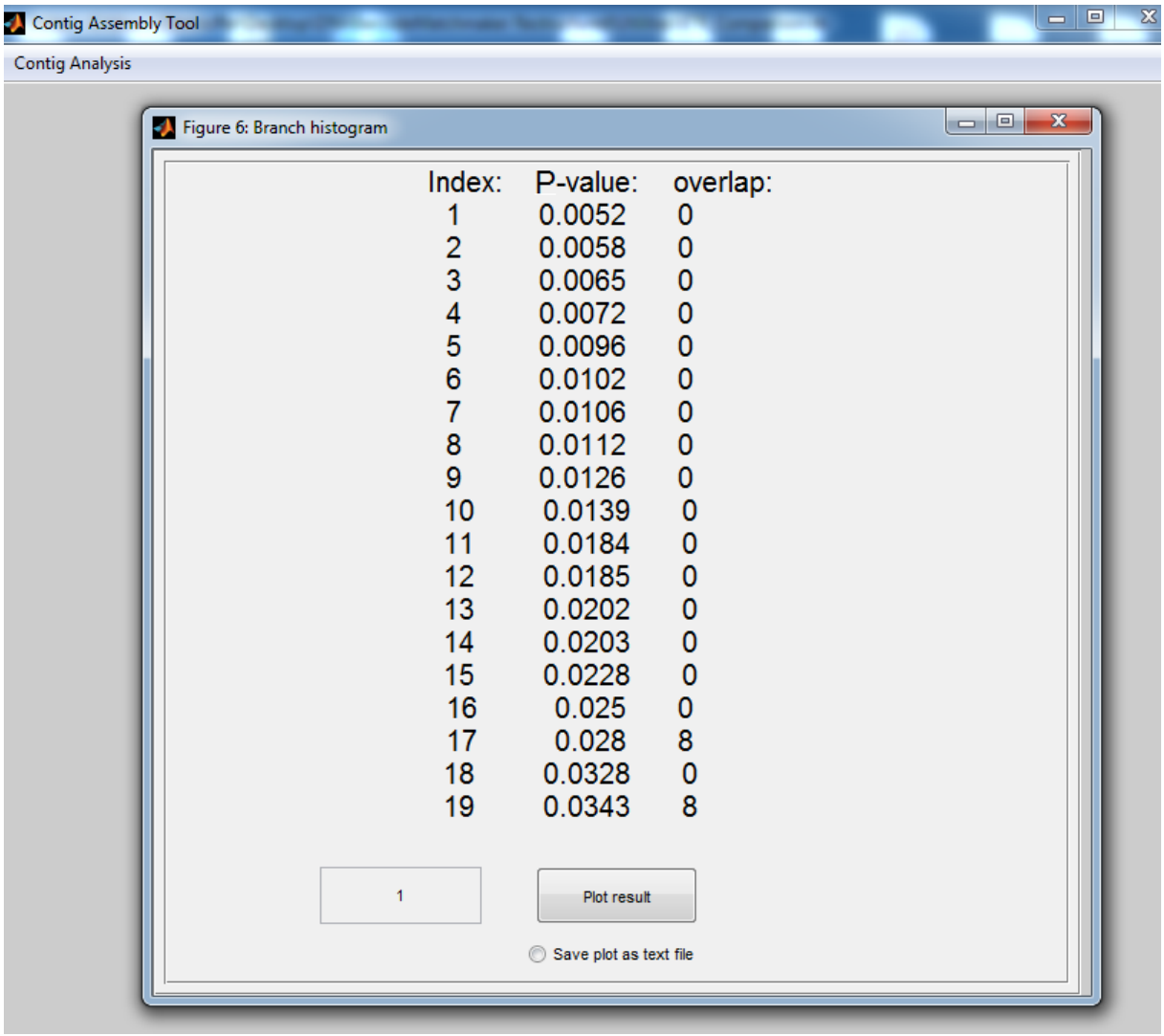


Figure 29. Result window for the Free Energy method version of the Contig Assembly software. Displays the 19 best ways of placing the contigs (within the limitations chosen in the menu found in Figure 27). The button labeled “Plot result”, plots the configuration specified by the number in the box to its left.

When the Zero Model option has been selected, the software starts matching the contigs according to the specified method (described in detail in section 7.3 and 7.4). After the calculations have been completed, the result window appears. For the Free Energy method, an example of the result window can be seen in Figure 29. The 19 best ways of placing the contigs is shown here, with corresponding P-value and number of pixels that overlaps between contigs. By pressing “Plot result”, the user can plot the result with index specified in the text box to the left of the button. The radio button “Save plot as text file”, can be checked if the software also should save the intensities of each contig barcode (as well as the template barcode) to a txt-file. Figure 30 is an example of the plot generated when pressing “Plot result” in Figure 29.

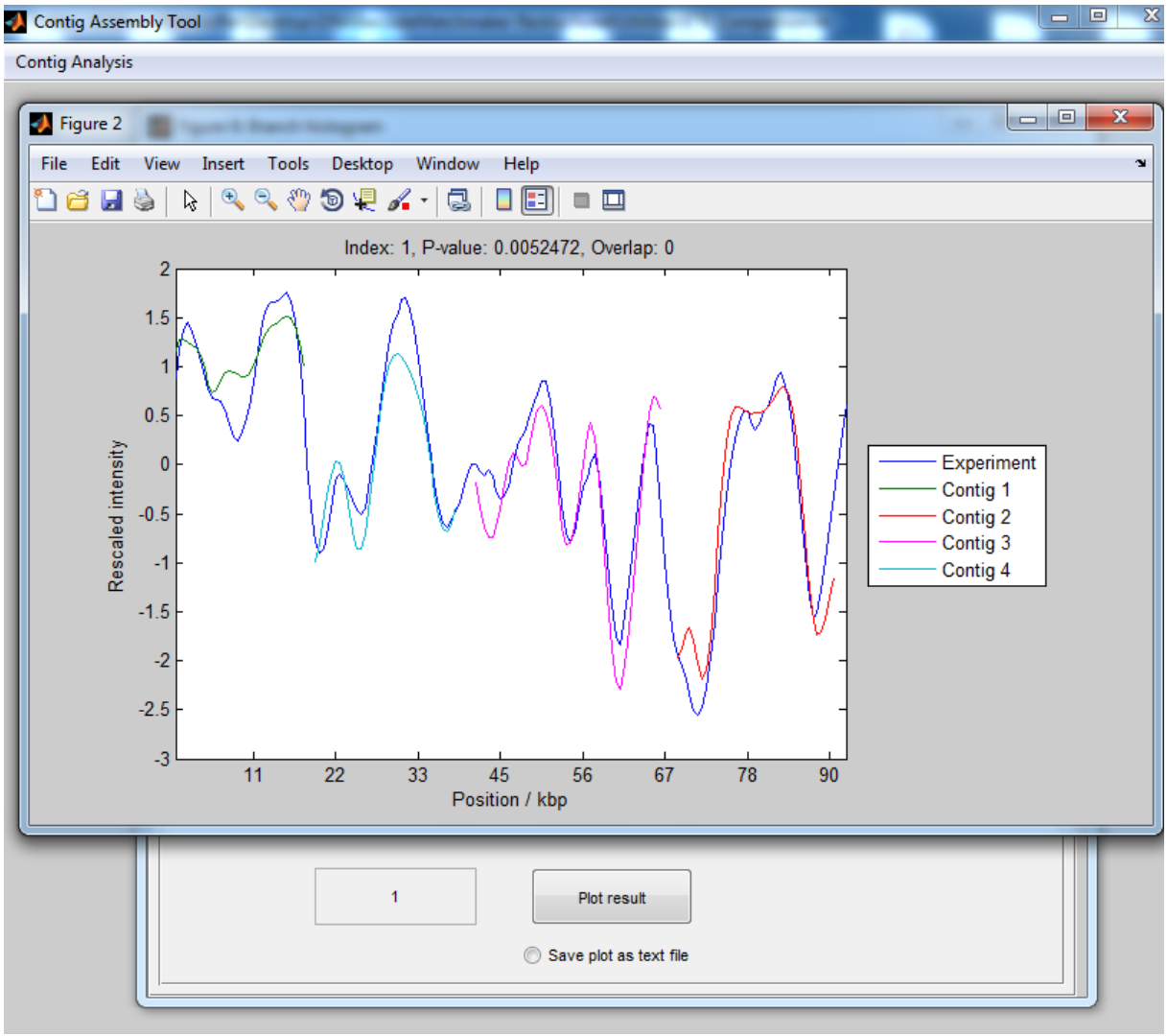
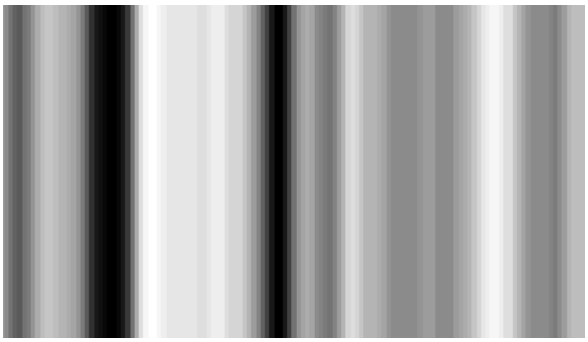


Figure 30. Plotted result for the Free Energy method version of the Contig Assembly software. Generated by pressing the “Plot result” button from Figure 29.

4. Appendix D – Popular Science Summary

In a world with ever increasing use of antibiotics, more and more bacteria are developing resistance to these medicines. If a person is infected with a strain of antibiotic resistant bacteria, then there may be no way for the doctors to help the patient and the patient might die as a worst case scenario. Furthermore, as with any other disease, the bacteria will spread to other people and thus taking their antibiotic resistance with them. This could potentially quickly escalate to a wave of death sweeping across the world, decimating the human population. Since medicine will not work on the disease, the only option is to isolate the patients that have been infected and stop the spreading that way. A prominent issue is to identify whom to isolate. With a new clever use of DNA barcodes, bacteria containing an antibiotic resistance gene can be identified.

A DNA barcode is, as the name suggests, a barcode containing information about a DNA molecule. A DNA molecule consists of two strands, built up by four bases: A, T, C and G. These bases form base-pairs (with the opposite strand on the DNA molecule), but can only form AT and CG. By learning the sequence of one of the strands, both strands have been identified. A base-pair is 0.3 nm wide, which makes it impossible to spot in a microscope. Because of this in traditional DNA sequencing, a very complicated (and slow) method is used in order to figure out the sequence of a given DNA molecule. A DNA barcode on the other hand do not go through this complicated process. Instead they are created by extracting the DNA molecule from a bacterium and then mixing it with two sets of dyes: one (non-fluorescent, “black”) that binds to AT base-pairs and one (fluorescent, “white”) that binds to CG base-pairs. After that, the molecule is stretched in a nano-channel (a very narrow channel) and is photographed. The photo now contains black and white stripes unique for that DNA molecule. Antibiotic resistance is a gene typically located in small circular parts of the DNA in bacteria called plasmids. By scanning the barcodes of plasmids obtained from a patient infected by antibiotic resistant bacteria and another patient, one can potentially determine if both of them are infected with the same bacterial strain.



A DNA barcode of a bacterium plasmid containing an antibiotic gene.

Even if the antibiotic resistance associated bacterial strain can be identified, some patient isolation procedure has to be put in place. There are two possible ways of isolating patients: Isolate everyone that comes to a hospital or isolate patients that do not respond to antibiotics. If everyone who was ill were to be isolated, the costs for hospitals would be enormous (and then we are not even considering the ethical dilemma of isolating people), and thus that is not an option, in general. That leaves us with isolating patients that do not respond to antibiotics. The problem with doing that is that it may already be too late since the disease could already have been spread to others (and these others could have passed it on to people they have met and so on). This would once again lead to isolating plenty of people, most of whom even would be completely healthy. It could take weeks for test results to confirm who has been infected, and during this time all that might have gotten infected have to remain in isolation. With DNA barcodes, this time can be reduced to hours, which makes this isolation approach acceptable. Thus the pandemic of deadly antibiotic resistant bacteria can potentially be avoided, and people can get back to their lives with only a minor inconvenience.