Comenius University Bratislava
Faculty of Mathematics, Physics and Informatics

Multiple Alignment and Visualization of Nanopore Sequencing Signals

Bachelor thesis

2018
Dávid Barbora

Comenius University Bratislava
Faculty of Mathematics, Physics and Informatics

Multiple Alignment and Visualization of Nanopore Sequencing Signals

Bachelor thesis

Study programme:	Computer Science
Study field:	2508 Informatics
Department:	Department of Computer Science
Supervisor:	doc. Mgr. Tomáš Vinař, PhD.

Bratislava, 2018
Dávid Barbora

Comenius University in Bratislava
Faculty of Mathematics, Physics and Informatics

THESIS ASSIGNMENT

Name and Surname: Dávid Barbora
Study programme: Computer Science (Single degree study, bachelor I. deg., full time form)
Field of Study: Computer Science, Informatics
Type of Thesis: Bachelor´s thesis
Language of Thesis: English
Secondary language: Slovak

Title: Multiple Alignment and Visualization of Nanopore Sequencing Signals

Annotation: Nanopore sequencing produces signals based on the underlying reference DNA. However, there is a large variability in these signals and at the same time, each position is typically read several times. The goal of the thesis is to develop practical multiple alignment methods for nanopore sequencing signals in order to align them to the reference DNA and to visualize the results.

Supervisor: doc. Mgr. Tomáš Vinař, PhD.
Department: FMFI.KAI - Department of Applied Informatics
Head of department: prof. Ing. Igor Farkaš, Dr.

Assigned: 31.10.2017

Approved: 31.10.2017 doc. RNDr. Daniel Olejár, PhD.
Guarantor of Study Programme

.....
Student

.....
Supervisor

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Dávid Barbora
Študijný program: informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: anglický
Sekundárny jazyk: slovenský

Názov: Multiple Alignment and Visualization of Nanopore Sequencing Signals
Viacnásobné zarovnávanie a vizualizácia signálov nanopórového sekvenovania

Anotácia: Výsledkom nanopórového sekvenovania sú postupnosti signálov vytvorené na základe referenčnej DNA. V týchto signáloch je vysoká variabilita a každá pozícia je typicky prečítaná viackrát. Cieľom práce je vytvoriť metódy pre praktické viacnásobné zarovnanie takýchto signálov a ich vizualizácia v kontexte referenčnej DNA.

Vedúci: doc. Mgr. Tomáš Vinař, PhD.
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: prof. Ing. Igor Farkaš, Dr.

Dátum zadania: 31.10.2017

Dátum schválenia: 31.10.2017 doc. RNDr. Daniel Olejár, PhD.
garant študijného programu

.....
študent

.....
vedúci práce

Acknowledgement: I would like to thank Tomáš Vinař for his guidance, patience and consultations. I would also like to thank Jakub Havelka for providing preprocessed data for testing.

Abstrakt

Výsledkom nanopórového sekvenovania sú postupnosti signálov vytvorené na základe referenčnej DNA. V týchto signáloch je vysoká variabilita a každá pozícia je typicky prečítaná viackrát. Štandardný postup pri analýze týchto dát je preložiť každý prečítaný signál na DNA sekvenciu. Tieto sekvencie sú vzájomne zarovnávné pre vyhladenie chýb vzniknutých pri preklade a vytvorenie výslednej sekvencie. Náš prístup je opačný. Najprv zarovnáme signály a vytvoríme z nich jeden ktorý bude preložený na DNA sekvenciu. V práci analyzujeme viaceré možnosti pre viacnásobné zarovnanie signálov s cieľom vytvoriť jeden signál ktorý vyprodukuje menej chýb pri preklade.

Kľúčové slová: nanopórové sekvenovanie, dynamic time warping, viacnásobné zarovnávanie

Abstract

Nanopore sequencing produces signals based on the underlying reference DNA. However, there is a large variability in these signals and at the same time, each position is typically read several times. The standard approach to analyze these data is to translate each signal read into the DNA sequence. These sequences are aligned to each other to fix mistakes introduced in process of translation and to produce consensus sequence. Our approach is different. At first we align signals and produce one signal and then we translate it to the DNA sequence. We analyze multiple approaches to multiple signal alignment with the goal of producing one signal that will result in fewer translation mistakes.

Keywords: nanopore sequencing, dynamic time warping, multiple alignment

Contents

List of Figures

Introduction

DNA sequencing is a great challenge of modern science. Ten years ago, it took several months and millions of dollars to sequence whole human genome. Modern technologies can do so in days at cost of thousands of dollars. These technologies are very precise, but the device is still very big and expensive.

In 2012, scientists at Oxford Nanopore Technologies developed the new portable device called MinION that can sequence DNA and costs only several thousands. This device allows us to sequence genomes at International space station or in extreme conditions on demand. MinION has, however, several issues. The main problem is precision of sequencing. This device measures a current flowing through nanopore with DNA molecule inside. The process of translating this data into DNA sequence is called basecalling and introduces a lot of errors due to disruptions in measured values.

Our goal is to get away from these disruptions by taking multiple reads of same DNA sequence and producing a single signal with fewer disruptions and higher quality. To do this we adapted approaches known from multiple alignment of DNA sequences along with dynamic time warping used to align signals.

In the first chapter, we introduce biological motivation and aspects of a MinION device along with standard data processing procedure for this data.

In the second chapter, we described known algorithms for sequence alignment, multiple sequence alignment, and alignment of signals.

In the third chapter, we describe our approaches to combine multiple sequence alignment with the alignment of signals and reconstruction of a signal from this alignment.

The fourth chapter shows visualizations we created to analyze the outcome of our approaches.

In the final chapter, we describe testing data and testing procedure to evaluate the quality of our approaches.

Chapter 1

DNA sequencing and MinION technology

In this chapter, we provide an introduction to DNA sequencing. We describe the new MinION sequencing technology, raw data provided by this technology and standard procedure to process such data.

1.1 Biological background

Every living organism has all the genetic information saved in DNA inside the core of every cell.

This genetic information helps us to find related species, discover genetic disorders or recognize bacteria.

The information inside DNA is composed of four nitrogenous bases adenine, guanine, cytosine, and thymine. We represent them with single letters A, C, T, G.

DNA sequencing is the process of determining the order of these bases within a DNA molecule. Various technologies to sequence DNA are known but most of them are big and expensive.

1.2 MinION technology

MinION (Fig. 1.1) is a new real-time sequencing technology developed by Oxford Nanopore Technologies. It is based on nanopores and it is highly portable.

A nanopore is a hole so small, that only single DNA strand fits in. MinION uses a protein nanopore set in an electrically resistant polymer membrane. An ionic current is passed through the nanopore by setting a voltage across this membrane. If an analyte passes through the pore or near its aperture, this event creates a characteristic

Figure 1.1: MinION device.

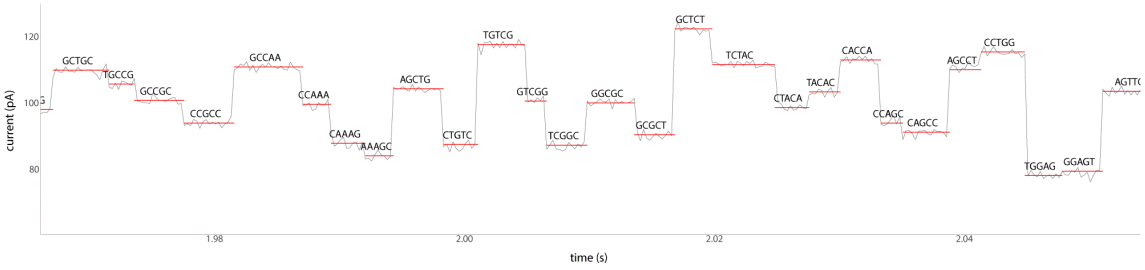

Figure 1.2: Raw signal generated by MinION device with corresponding 5-mers [1].

disruption in the current. Measurement of that current makes it possible to identify the molecule in question [5].

1.2.1 Processing of MinION data

Raw data produced by MinION device consist of current measures in picoamperes. Each measured value corresponds to five or six bases passing nanopore called 5-mers or 6-mers (Fig. 1.2). Each k-mer is measured several times as passing through a nanopore.

The MinION device typically produces many reads covering different parts of processed DNA molecule. The process of translating current measures into a sequence of bases (letters A, C, G, T) is called basecalling. It is slow process usually based on hidden Markov model or recurrent neural networks.

Regular approach to analyzing MinION data is to basecall all reads and find their order and overlap by standard sequence aligning algorithms.

Chapter 2

Sequence Alignment

Alignment is the task of locating equivalent regions of two or more sequences to find their similarity. The large similarity often mean the same biological function or common ancestor in the evolutionary history. In this chapter, we formalize the problem of alignment and present standard algorithms for solving this problem.

2.1 Pairwise alignment

Definition 2.1.1 (Alignment)**.** Let $u = u_1 \dots u_n$, $v = v_1 \dots v_m$ be two sequences where $u_i, v_i \in \{A, C, T, G\}$ and M be a matrix

$$M = \begin{pmatrix} M_{1,1} & M_{1,2} & \dots & M_{1,k} \\ M_{2,1} & M_{2,2} & \dots & M_{2,k} \end{pmatrix}$$

We call M an alignment of u and v if:

1. $\forall i, j : M_{i,j} \in \{A, C, T, G, -\}$
2. $M_{1,1}M_{1,2} \dots M_{1,k}$ is a word created by inserting dashes into u
3. $M_{2,1}M_{2,2} \dots M_{2,k}$ is a word created by inserting dashes into v
4. No column contains two dashes

For example

$$\begin{pmatrix} - & G & T & A & C & G & T & C & C & T & A & A \\ T & G & T & A & C & G & C & C & - & T & - & - \end{pmatrix}$$

is one of many possible alignments of sequences $GTACGTCCTAA$ and $TGTACGCCT$. Another alignment of these sequences could be

$$\begin{pmatrix} - & - & - & - & G & T & A & C & G & T & C & C & T & A & A \\ T & G & T & A & C & G & C & C & - & - & - & - & T & - & - \end{pmatrix}$$

To find the best alignment (with most of the similarities), we define a scoring system which assigns a numeric value to the alignment.

A basic scoring system can be $+1$ for each matching column, -1 for every other column.

Definition 2.1.2 (Global alignment)**.** Given two sequences and a scoring system, the global alignment is the alignment with the highest score in the scoring system.

Sometimes, it is better to search for the most similar part of the two sequences. We call this approach the local alignment

Definition 2.1.3 (Local alignment)**.** Given two sequences u, v and a scoring system. Let u' and v' be substrings of u and v such that their global alignment has the highest score among all substrings of u and v. Local alignment of u and v is the global alignment of u' and v'.

2.2 Needleman-Wunsch algorithm

Needleman-Wunsch algorithm [7] for computing global alignment is based on dynamic programming.

Let $u = u_1 \dots u_n$, $v = v_1 \dots v_n$ be two sequences. We construct matrix $A[n, m]$ where $A[i, j]$ is the global alignment score of $u_1 \dots u_i$ and $v_1 \dots v_j$.

Let $s(i, j)$ be score of alignment of u_i and v_j (1 if they are equal, -1 otherwise). Matrix A is constructed dynamically row by row, where $A[i, j]$ is computed as the maximum of three possibilities:

1. $A[i-1, j-1] + s(i, j)$
2. $A[i-1, j] - 1$
3. $A[i, j-1] - 1$

In the first case, we align u_i to v_j as a match or mismatch. In the second case, we insert dash into the first sequence. In the third case, we insert dash into the second sequence.

The final alignment score is $A[n, m]$ and the alignment can be reconstructed by finding the path of computation from $A[n, m]$ to $A[1, 1]$. This path can be simply found by following the highest values out of the three possibilities used to compute the values in matrix A.

Needleman-Wunsch algorithm can be easily modified to find the local alignment in two steps. First, we add the fourth case:

4. 0

This allows us to change the starting point of the alignment if doing so leads to a better solution. Second, the final score is the maximum value in the matrix, which allows us to change ending points of the alignment. Reconstruction of the alignment is done by following the path from the maximum value in the matrix to the closest 0, where the local alignment begins.

2.3 Multiple alignment

If we have a large dataset of evolutionarily related DNA sequences that share a common ancestor, we might find something about such ancestor, if we look at their overall similarity.

Definition 2.3.1 (Multiple alignment)**.** Let $u_1 = u_{1,1} \dots u_{1,m_1}, \dots, u_n = u_{n,1} \dots u_{n,m_n}$ be n sequences where $u_{i,j} \in \{A, C, T, G\}$ and M be a matrix

$$M = \begin{pmatrix} M_{1,1}\ M_{1,2} \dots\ M_{1,k} \\ M_{2,1}\ M_{2,2} \dots\ M_{2,k} \\ \vdots \qquad\qquad \vdots \\ M_{n,1}\ M_{n,2} \dots\ M_{n,k} \end{pmatrix}$$

We call M an alignment of $u_1, \dots, u_n$ if:

1. $\forall i, j : M_{i,j} \in \{A, C, T, G, -\}$
2. $M_{i,1} M_{i,2} \dots M_{i,k}$ is a word created by inserting dashes into u_i
3. No column contains n dashes

We can see that the alignment of two sequences is a special case of a multiple alignment.

2.3.1 Dynamic programming

We can modify Needleman-Wunsch algorithm to solve this problem. Matrix A would become n-dimensional and the time complexity of computing such matrix would be $O(\prod_{i=1}^{n} m_i)$. Matrix A can still be constructed dynamically row by row, where $A[i_1, ...i_n]$ is computed as the maximum of $n + 1$ possibilities:

1. $A[i_1 - 1, i_2 - 1, \dots, i_n - 1] + s(i, j)$
2. $A[i_1 - 1, i_2, i_3, \dots, i_n] - 1$

k+1. $A[i_1, \dots, i_k - 1, \dots, i_n] - 1$

n+1. $A[i_1, i_2, i_3, \ldots, i_n - 1] - 1$

To find the global optimum for n sequences has been shown to be an NP-complete problem [10].

2.3.2 Heuristics

Since DNA sequences are usually large, exponential time complexity is unacceptable. Several heuristic algorithms were developed, with different time complexities and accuracy.

Progressive alignment

The most common heuristic approach to multiple alignment is a progressive technique [3]. Progressive alignment builds up the final alignment by combining pairwise alignments. A binary tree called a guide tree with sequences as leaves decide an order of pairwise alignments. The final alignment is built by following that tree from leaves to root. Such tree can be composed by finding most similar pairs among all possible pairwise alignments resulting into a quadratic number of alignments. The problematic part of this approach is aligning two alignments which can be done in many various ways [11].

Iterative methods

Another method is to construct the alignment by adding sequences one by one to the final alignment. This way, we only need a linear number of alignments, but we are aligning one sequence to an alignment, which can again be done in various ways. Efficiency is improved at the cost of accuracy [9].

Chapter 3

Squiggle alignment

In this chapter, we describe our work. We took a different approach from standard procedure described in 1.2.1. To compose DNA sequence, we took squiggles that correspond to the same part of the reference sequence and aligned them together to produce one signal that will be later basecalled. We designed several ways of multiple squiggles alignment to generate the best signal according to similarity of sequence from basecalling and reference sequence.

3.1 Squiggles and preprocessing

MinION device produces sequences of measured values of the current passing nanopores. We call these sequences squiggles. Typically, we have multiple squiggles covering each part of the DNA. The raw signal contains values in pA between 0 and several hundred. Each pore produces slightly different data. Current values are shifted and scaled by some constant.

The best way to hide these differences is to scale the sequence so that the resulting mean value is 0 and the standard deviation is one [2].

3.2 Squiggle alignment with dynamic time warping

To align two squiggles, we use the method called dynamic time warping (DTW). DTW is widely used in audio processing and speech recognition [6]. It uses a similar approach as the Needleman-Wunsch algorithm for sequence alignment, but does not specify the score of a match or a mismatch, but instead assigns a cost function for any pair of values.

The cost function $c(i, j)$ which assigns a cost for aligning value i to value j. It can be, for example, distance $abs(i - j)$ or square of distance $abs(i - j)^2$.

By evaluating a cost function for two sequences u and v, we calculate the cost

matrix C, where $C[i,j] = c(u_i, v_j)$. The best alignment of these two sequences can be represented as a continuous path from $C[1,1]$ to $C[n,m]$, with the lowest sum of costs (Fig. 3.1). We call such a path a warping path and the sum of costs will be the cost of the alignment. The cost of the alignment is also indicative of the similarity of those sequences.

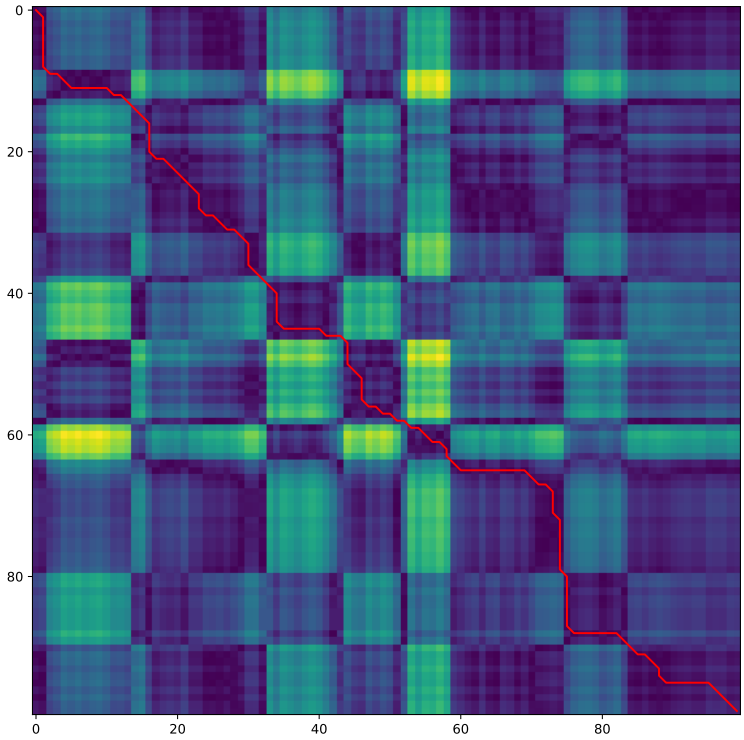

Figure 3.1: Warping path in matrix C.

The warping path can be represented as a list of coordinates from matrix C.

To find the best path, we will not compute matrix C containing the costs of aligning single values. Instead, we will calculate matrix A, where $A[i,j]$ is the cost of the alignment of sequences $u_1 \dots u_i$ and $v_1 \dots v_j$, similarly as we did in the Needleman-Wunsh algorithm. In particular, $A[i,j]$ will be computed as $min(A[i-1,j], A[i,j-1], A[i-1,j-1]) + c(u_i, v_j)$. To find the warping path in A, we follow the lowest values from $A[n,m]$ to $A[1,1]$.

The time complexity of this algorithm is quadratic, but can be easily reduced to almost linear by restricting the area where the warping path is allowed to pass to a band with a constant width (Fig. 3.2). We will define $c(i,j) = \infty$ for all i,j not in this band. This way, the values that are not computed, will not affect the final value.

By checking the coordinates of a chosen point while constructing the warping path, we can find out if the chosen band covers the whole warping path. When the band is not wide enough, we will restart the computation with a band that will be twice as wide [8].

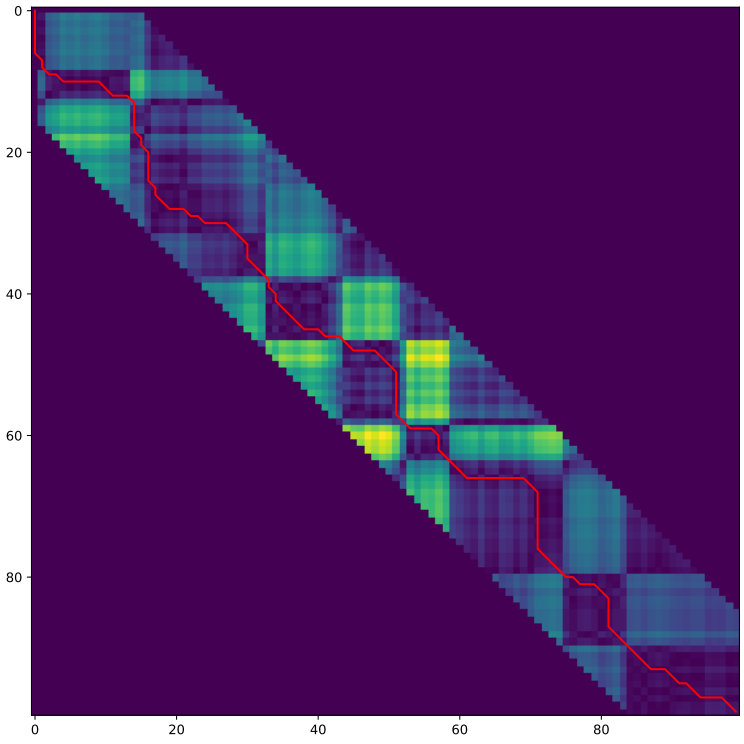

Figure 3.2: Band restricted warping path in matrix C.

3.3 Signal reconstruction from the warping path

To facilitate an iterative alignment of the squiggles, we need to be able to create one signal out of two. We will use the warping path created by DTW. Although the signals are very similar, alignment still contains some insertions in both squiggles (Fig. 3.3).

When aligning DNA sequences, we had three possibilities. Match, mismatch, or aligned to a dash. This time, we have a pairing of points. Each point from the first sequence has the corresponding point in the second sequence and each point from the second sequence has the corresponding point in the first sequence. This also means that sometimes one point from the first sequence corresponds to multiple points from the second sequence or vice versa.

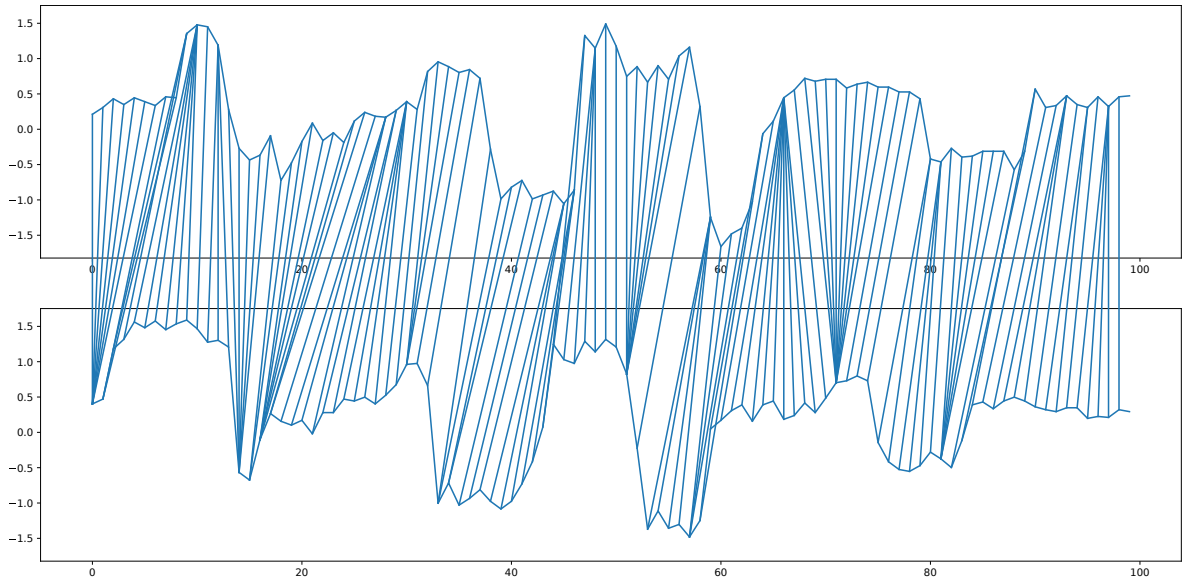

Figure 3.3: Pairing of points created by DTW.

We considered multiple ways to solve this problem.

Complete alignment

The first approach is to calculate an average of each pair of points in alignment and concatenate them to the final signal. The signal we create in this way is slightly longer.

Alignment to sequence

The second approach is to take the first sequence as leading and to construct the final signal by calculating an average of a point from this sequence and all points aligned to it, in every point of this sequence. This approach puts a high weight on the first sequence. If some part of the signal from the second sequence is missing in the first sequence, it will never appear in the alignment. The same holds if some parts of the first signal is unusually long. A single point from the second signal will align to it, and the whole length of the signal will stay there. An advantage of this approach is, that the resulting signal looks exactly like a signal that MinION produces and the basecaller expect.

3.4 Multiple alignment

To align multiple signals, we have chosen the iterative method, as signals are much longer than the DNA sequences, and the iterative method is faster. However if we would just add the sequences one-by-one and each time would generate a resulting signal, weight of the squiggles added later will be higher. We have tried multiple

approaches to solve this problem.

Simple average

The first approach was to calculate the weighted average when aligning the i-th signal to the consensus of $i-1$ already aligned signals. The result is that at each point, every sequence has the same weight in consensus.

Average with length adjustment

In the second approach, we remembered how many points were aligned from all sequences to each point of the consensus. We used this information when reconstructing the final signal. For each point, we calculated an average number of points aligned to it and adjusted its weight in the final sequence accordingly.

With these different ways of reconstructing the final sequence, we can afford to take the full alignment each time. While it will result in much longer signal than expected, it will be later reduced by realigning all sequences to it and counting points aligned to each point in it.

Chapter 4

Visualization

To analyze the problems of different approaches we designed multiple visualizations.

4.1 Raw data

The simplest but often very helpful was to directly plot raw signals of all squiggles we were aligning. As all our approaches start with mean value and standard deviation preprocessing, we used these preprocessed data for visualization. We can see that this can reveal unexpected defects in data (Fig. 4.1).

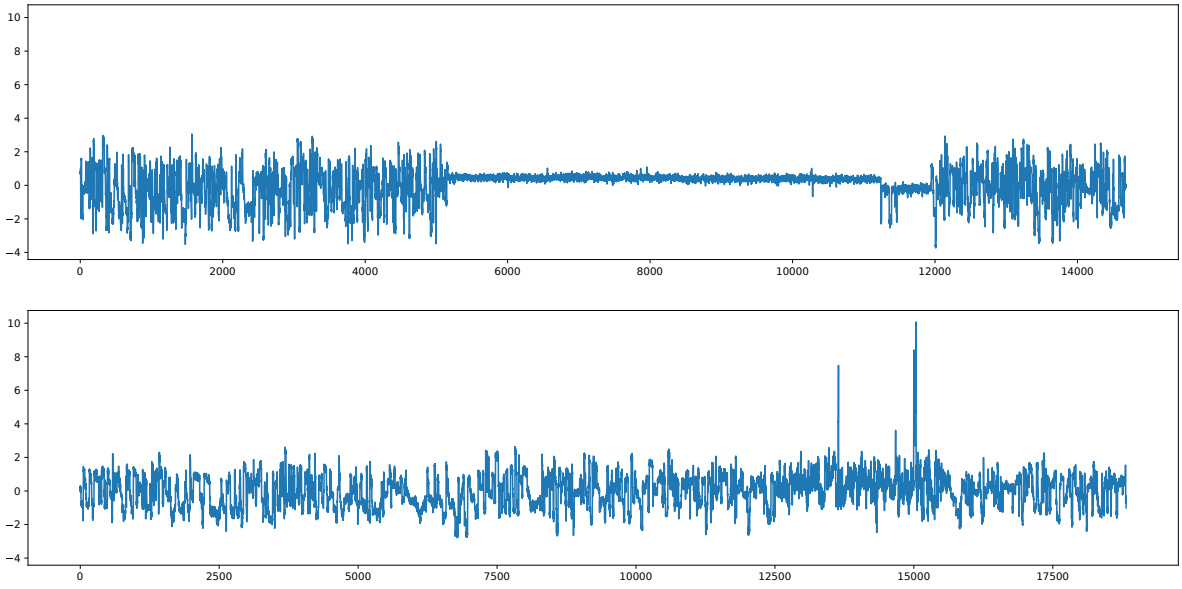

Figure 4.1: Defect data found by simple visualization.

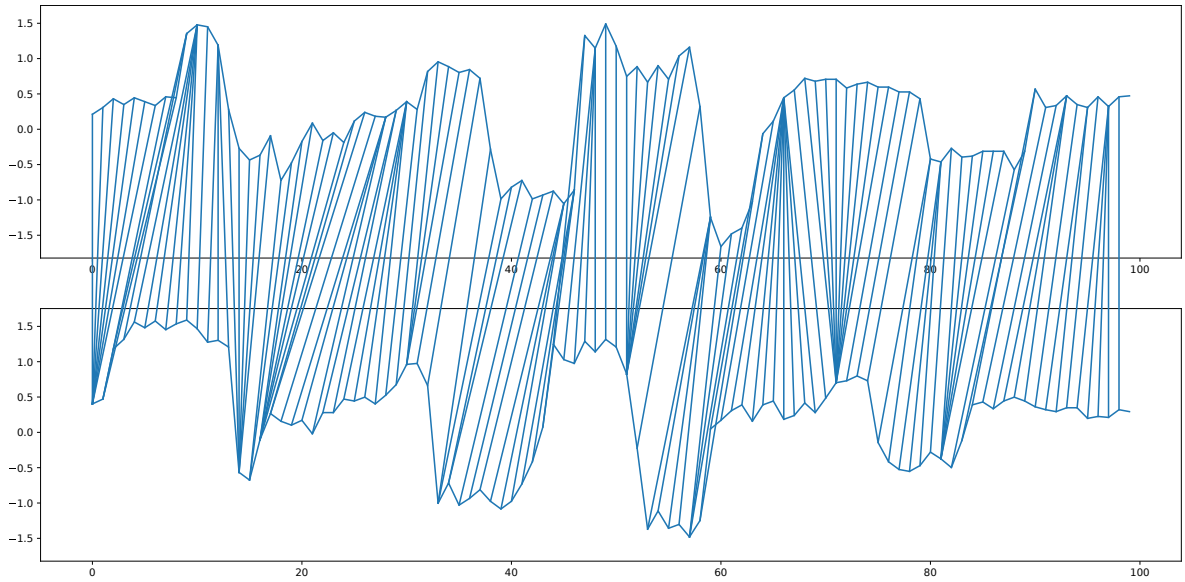

Figure 4.2: Pairing of points created by DTW.

4.2 Alignment

To visualize single alignment of signals, we printed two signals vertically side by side and connected aligned points between those plots (Fig. 4.2).

We also extended this visualization for iterative alignment method. After each iteration of iterative alignment, we can generate the resulting signal and align it with the previous resulting signal to visualize changes made by last added squiggle. This way we can plot all interim results underneath and print lines between successive pairs (Fig. 4.3).

To visualize all aligned squiggles in context of their consensus signal, we took the consensus signal as leading and aligned each squiggle to it. This way we will loose omitted parts in squiggles, but all squiggles will have the same length and can be printed over each other with consensus signal (red) on top of them (Fig. 4.4).

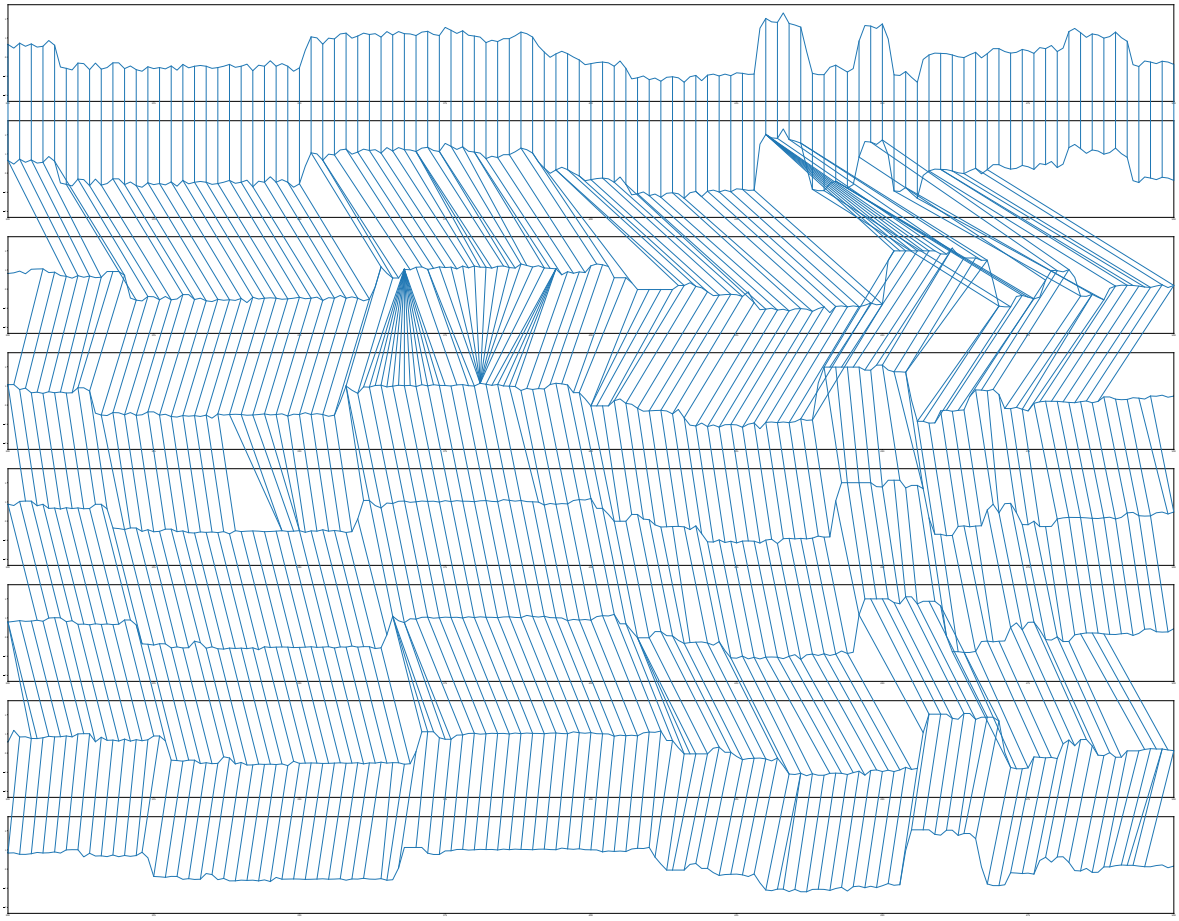

Figure 4.3: Insertions and deletions in process of iterative alignment.

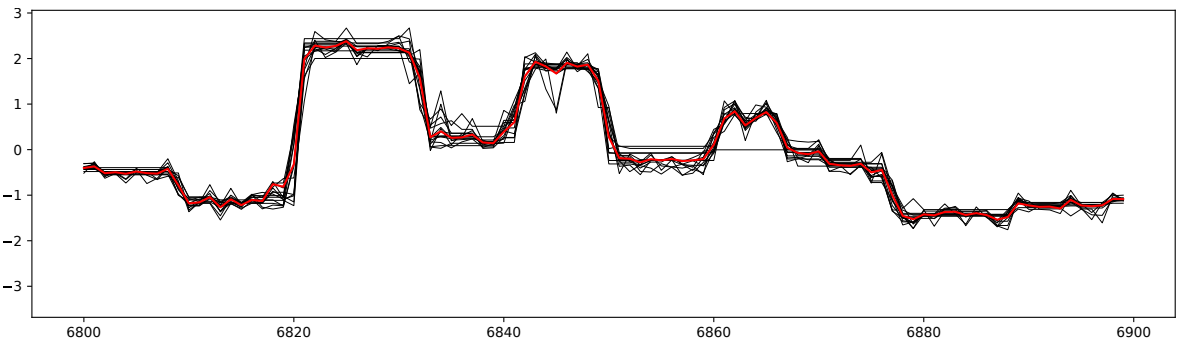

Figure 4.4: Aligned squiggles in context of consensus signal.

Chapter 5

Experiments and results

For all experiments we used data from *NA12878 Oxford Nanopore Human Reference Datasets* [4]. This dataset provides a lot of basecalled reads in fast5 format aligned to reference DNA.

For our experiments, we used 40 blocks of 1000 bases with corresponding parts of squiggles provided by bachelor thesis by Jakub Havelka [2].

To analyze the quality of our consensus signal, we used basecaller *ONT Albacore Sequencing Pipeline Software (version 2.2.4)*. At first, we basecalled each aligned squiggle and a consensus signal. Then for each DNA sequence produced, we calculated alignment score by aligning it to the corresponding part of the reference sequence using Needleman-Wunsch algorithm described in 2.2.

5.1 Results

We tested multiple methods of multiple alignments and signal reconstruction on all 40 datasets. Each graph shows alignment score of squiggles (blue) and consensus signal (red). Each column in graphs represents one dataset.

Figure 5.1 shows results of consensus signal produced by *aligning to sequence* and signal reconstruction with *simple average* after each alignment. We can see that consensus signal produced in some cases slightly better results than each squiggle but in most cases resulted in average quality.

Figure 5.2 shows results of consensus signal produced by *aligning to sequence* and final signal reconstruction by *average with length adjustment*. This Approach produced best results with better alignment score in more than 25% of datasets.

Figure 5.3 shows results of consensus signal produced by *aligning to sequence* and signal reconstruction by *average with length adjustment* after each alignment. This approach led to bad results in all cases.

Figure 5.4 shows results of consensus signal produced by *complete alignment* and

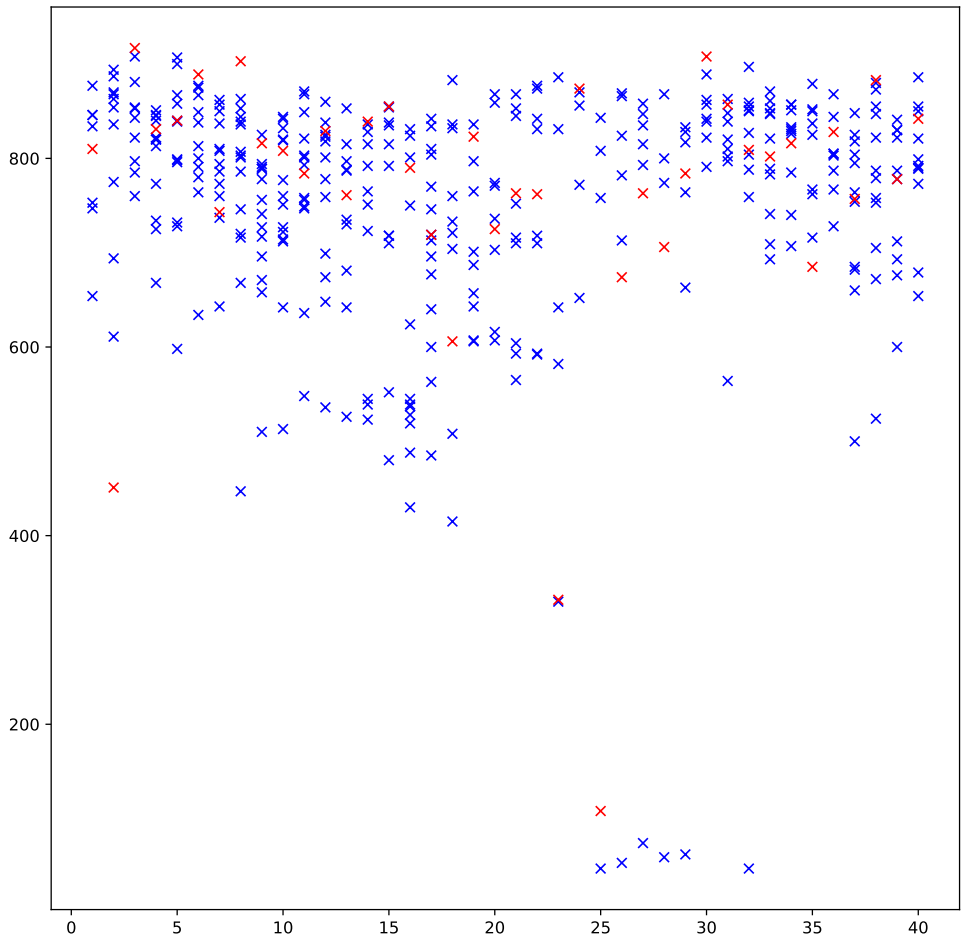

Figure 5.1: *Aligning to sequence* and signal reconstruction with *simple average* after each alignment.

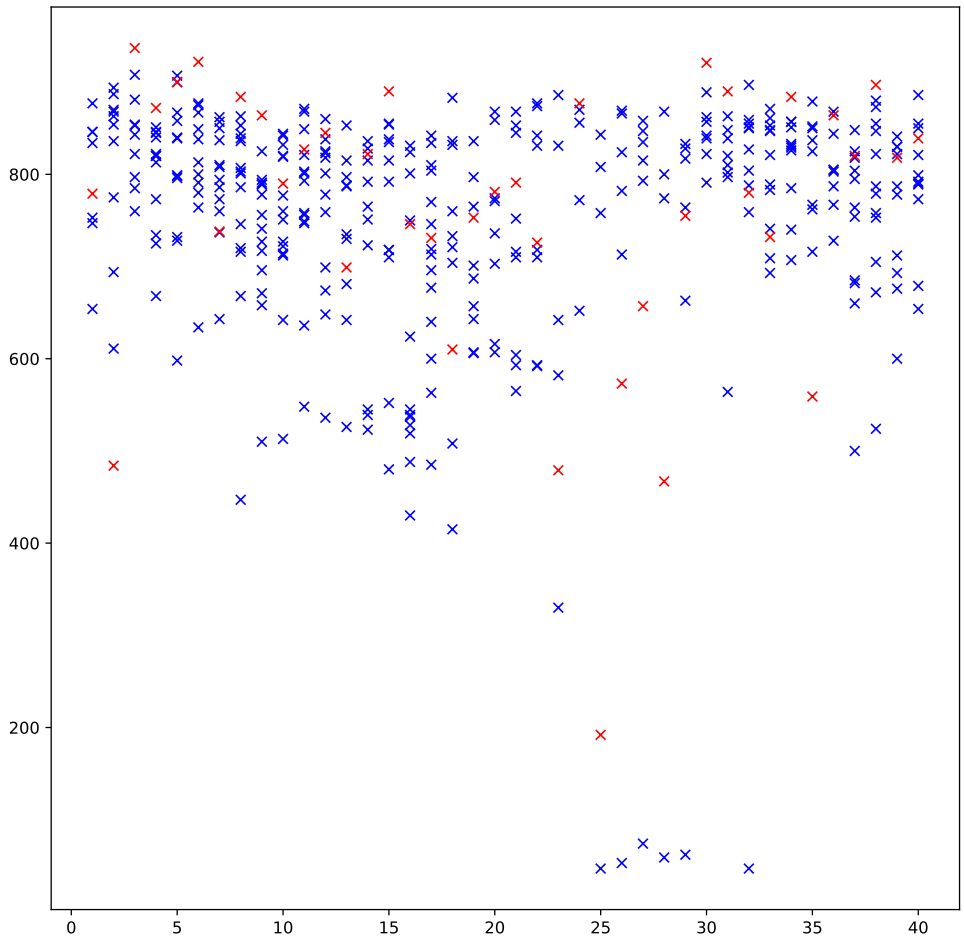

Figure 5.2: *Aligning to sequence* and final signal reconstruction by *average with length adjustment*.

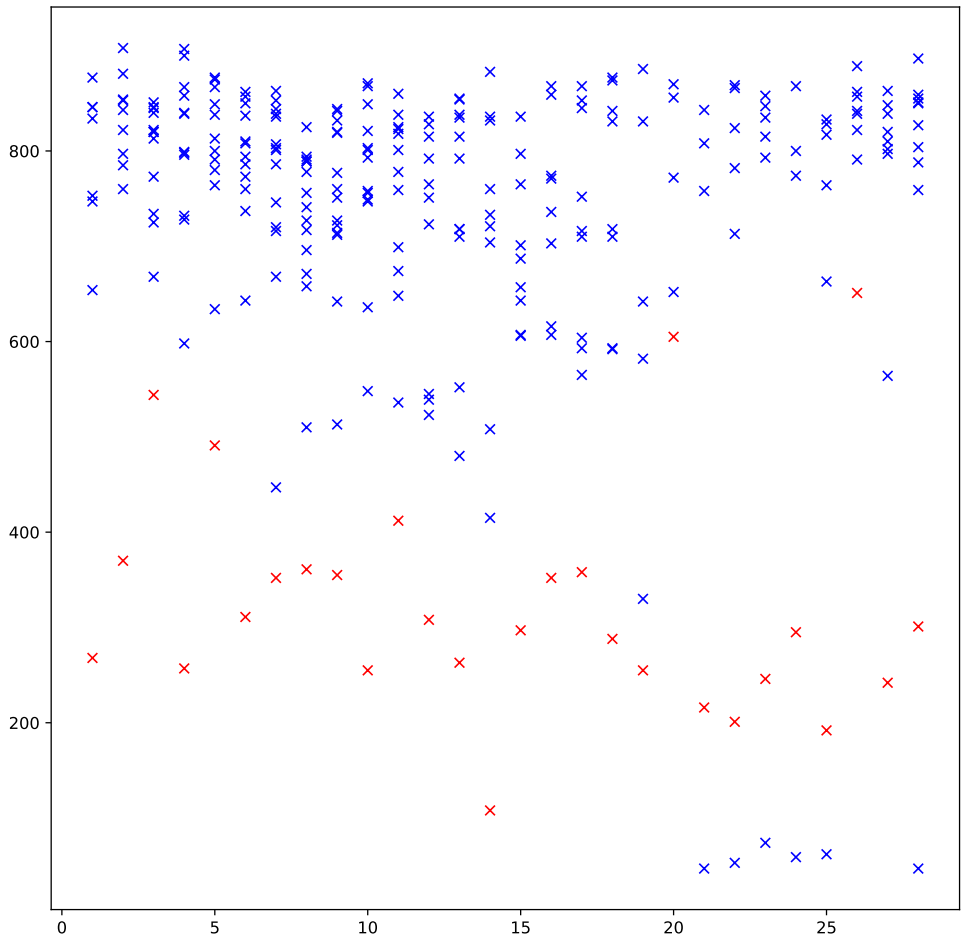

Figure 5.3: *Aligning to sequence* and signal reconstruction by *average with length adjustment* after each alignment.

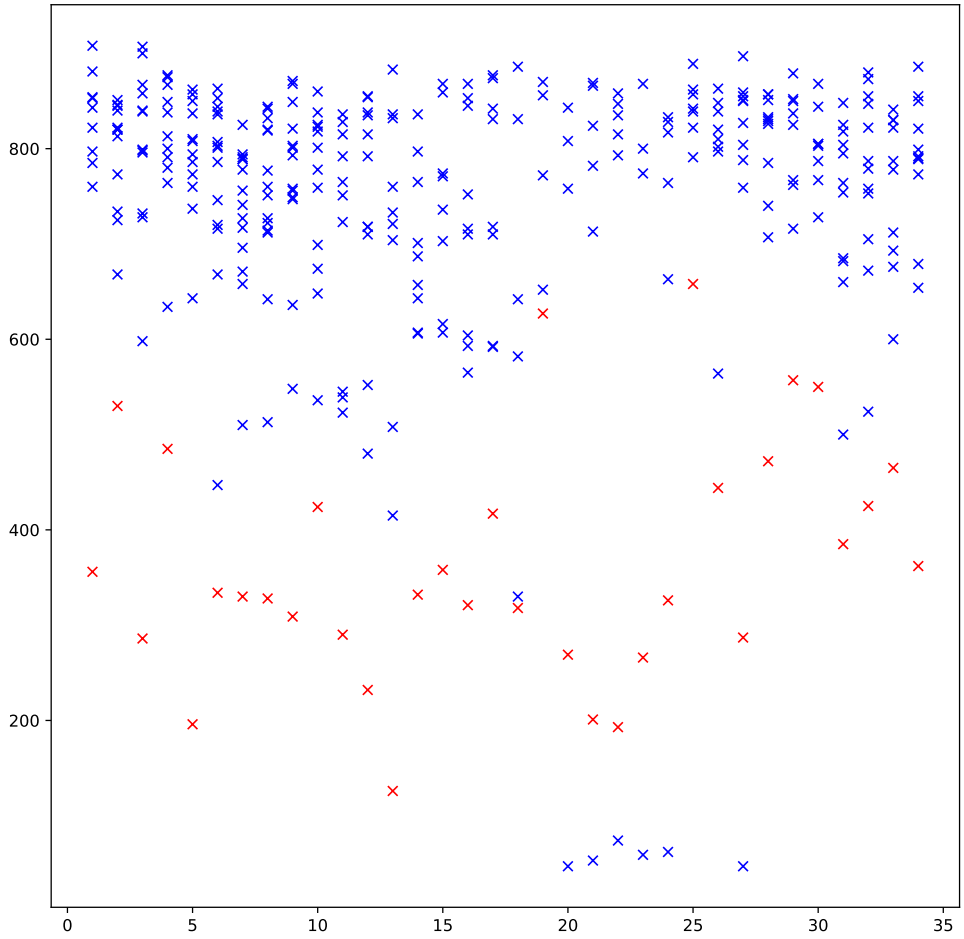

Figure 5.4: *Complete alignment* and signal reconstruction by *simple average*.

signal reconstruction by *simple average*. In this case, in some datasets resulting signal did not even pass quality filter of basecaller.

Figure 5.5 shows results of consensus signal produced by *complete alignment* and final signal reconstruction by realigning all squiggles to it by *aligning to sequence* and final signal reconstruction by *average with length adjustment*.

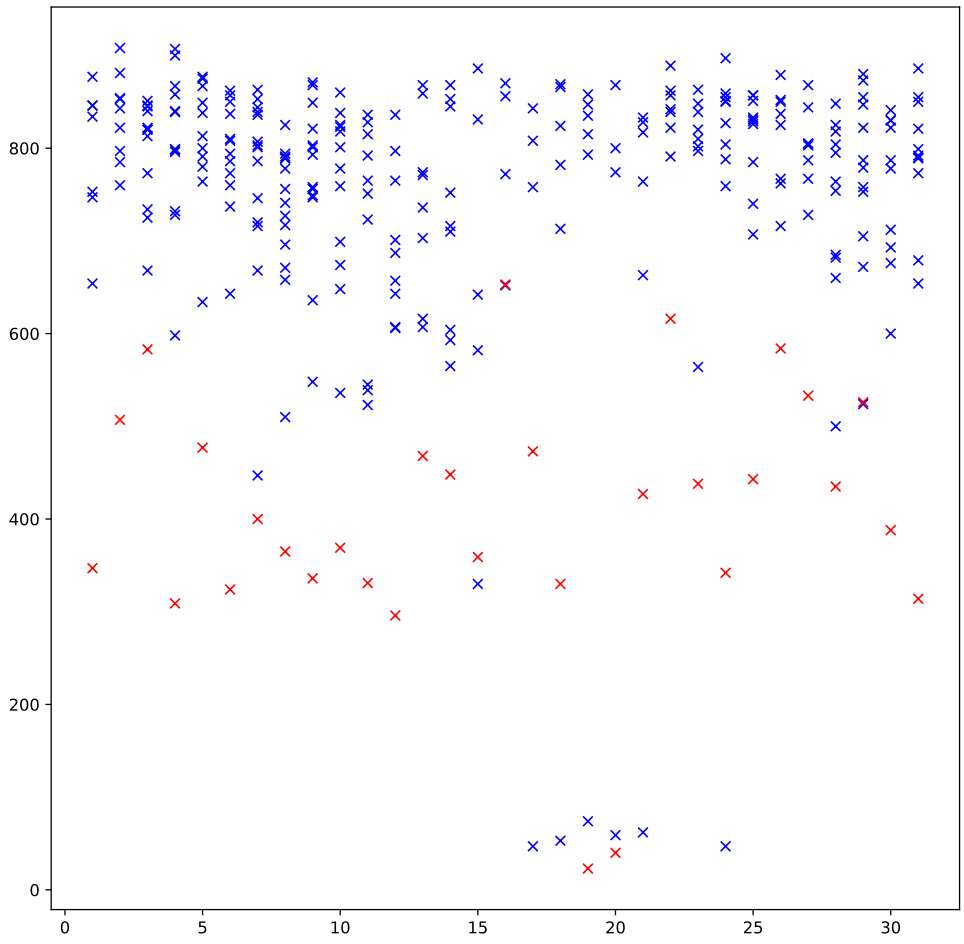

Figure 5.5: *Complete alignment* and final signal reconstruction by realigning all squiggles to it.

Conclusion

The most common approach to the processing of MinION data is to basecall and align to reference. The process of basealling is slow and introduces multiple errors. Our goal was to develop methods of multiple signal alignment as a way to produce a better signal that will result in the better outcome of basecalling.

To align two signals we adapted and modified dynamic time warping algorithm known from speech recognition. A special challenge was to adapt multiple sequence alignment techniques and modify them for use with signals and dynamic time warping.

We developed two ways of signal reconstruction from alignment produced by dynamic time warping. First one (*alignment to sequence*) was to take one sequence as leading and in each point of this sequence calculate an average of all points aligned to it. This approach does not change the overall structure of signal but smoothes the signal. The second approach (*complete alignment*) was to concatenate average of each pair of points produced by alignment. This approach results in a slightly longer signal that contains more information but does not terminate outlying values. It is practically unusable for multiple alignment as it results in longer signal after each alignment and basecaller does not understand such signal.

According to our tests, best results are achieved by combining *aligning to sequence* with final signal reconstruction by taking *average with length adjustment*. This approach resulted in a better signal in 27% of test cases.

The next step of MinION signals alignment would be to extend our methods for local alignment and produce one signal covering whole DNA sequence out of all squiggles. This could save time spent for basecalling all squiggles and also might result into more precise DNA sequence.

Bibliography

[1] C de Lannoy, D de Ridder, and J Risse. The long reads ahead: de novo genome assembly using the minion [version 2; referees: 2 approved]. *F1000Research*, 6(1083), 2017.

[2] Jakub Havelka. Alignment of nanopore sequencing squiggles. 2017.

[3] Des Higgins. Multiple sequence alignment. In *Genetic Databases*, pages 165–183. Elsevier, 1997.

[4] Miten Jain, Sergey Koren, Karen H Miga, Josh Quick, Arthur C Rand, Thomas A Sasani, John R Tyson, Andrew D Beggs, Alexander T Dilthey, Ian T Fiddes, et al. Nanopore sequencing and assembly of a human genome with ultra-long reads. *Nature biotechnology*, 2018.

[5] Alexander S Mikheyev and Mandy MY Tin. A first look at the oxford nanopore minion sequencer. *Molecular ecology resources*, 14(6):1097–1102, 2014.

[6] Meinard Müller. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84, 2007.

[7] Saul B Needleman and Christian D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, 1970.

[8] Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, 26(1):43–49, 1978.

[9] Iain M Wallace, O'Sullivan Orla, and Desmond G Higgins. Evaluation of iterative alignment algorithms for multiple alignment. *Bioinformatics*, 21(8):1408–1414, 2004.

[10] Lusheng Wang and Tao Jiang. On the complexity of multiple sequence alignment. *Journal of computational biology*, 1(4):337–348, 1994.

[11] Marketa J Zvelebil and Jeremy O Baum. *Understanding bioinformatics*. Garland Science, 2007.