PhD confirmation report on

NanoSplicer: alternative splicing analysis using Oxford nanopore sequencing

Candidate:

Yupei You

University of Melbourne

Supervisors:

Dr. Heejung Shim

Dr. Mike Clark

Advisory Committee Chair:

Prof. David Balding

Abstract

Alternative splicing is an essential mechanism that enables a single gene to produce multiple mRNA products (called isoforms). Oxford Nanopore sequencing produces long reads that have natural advantages for characterising isoforms. Alternative splicing can not only add or skip entire exons, but can also vary exon boundaries by selecting different nucleotides as splice sites. However, accurately identifying the latter is challenging for nanopore sequencing, as exon boundaries are often unclear due to the high error rate. One existing solution is polishing nanopore reads with short reads. While feasible, this approach requires both short and long reads, which adds considerable expense. Furthermore, isoform-distinguishing short reads are not always available (e.g. in 10X scRNAseq). Therefore, a method that could accurately identify exon boundaries solely from nanopore data would have numerous advantages.

I developed a method called "NanoSplicer" to identify splice sites using only nanopore sequencing data. Nanopore sequencing records changes in electrical current when a DNA or RNA strand is traversing through a pore. This raw signal (known as a squiggle) is then basecalled by computational methods, but this process is error-prone. Instead of looking at the basecalled sequences only, I also used the squiggle to identify splice sites. I tested my method using synthetic mRNAs with known splice sites, demonstrating our method can correctly assign 90.8% of the squiggles overlapping exon junctions to the correct splice sites. I conclude that using squiggles is a promising approach for accurately identifying splice sites.

Contents

Literature review

1.1 The role of alternative splicing in biology

Alternative splicing is an essential mechanism in eukaryotic cells that increases the diversity of transcripts by enabling a single gene to produce multiple mRNA products. It plays a vital role in differential gene expression between cell types, which is critical for cell differentiation, development and reprogramming, as well as tissue remodelling[1, 2, 3]. Abnormalities in alternative splicing has also been linked to human diseases[4, 5, 6, 7, 8].

Alternative splicing is known to happen on more than half of human genes[9]. In this process, different parts of a precursor mRNA will be included or excluded to form different mature mRNAs (called isoforms), which may have different biological properties or functions. Hence, it is critical to identify and quantify mRNA isoforms to study the function of genes.

1.2 Short-read sequencing in transcriptomics

The advent of next generation sequencing technologies revolutionized the field of transcriptomics, especially the quantification of gene expression and identification of novel transcripts. The significant development was high-throughput short-read sequencing (hereafter referred to as "short-read" sequencing, e.g. Illumina), which came to dominate the sequencing market. This technology improved sequencing efficiency by sequencing millions of short fragments of cDNA in parallel (the outputs are known as reads). Methods using short-read sequencing for transcriptome analysis (RNA-seq) are well established but have several disadvantages. Mature mRNA averages ~2 kb in length in humans[10]. The

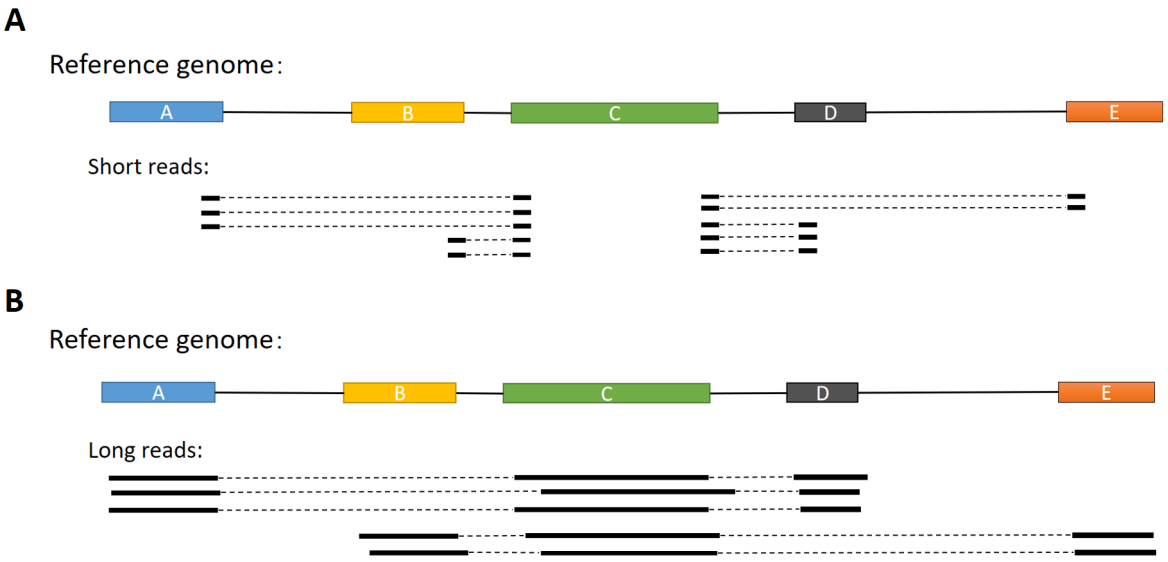

Figure 1.1: Comparison of transcriptome analysis using short reads and long reads
In both figure A and B, colored boxes in reference sequence present exons and the black lines connecting the boxes present introns. The lines below the reference are the mapped junction short reads (A) or long reads (B).

read length of RNA-seq (∼150 nt) is much below the transcript length, so that computational reconstruction of full-length transcripts is required (Figure 1.1A). Steijger et al. conducted a benchmarking study[11] evaluating 14 computational methods and showed the methods have the potential to discover novel exons and exon junctions but are not robust enough for full-transcript isoform reconstruction even with increased read depth. This is because each short read has information on only a small part of a isoform, and isoforms from the same gene often share a large amount of sequence[12].

1.3 Oxford Nanopore Technology

The first idea of using nanopore-based approaches for detecting nucleotide sequence emerged in the 90s[13, 14]. In 2012, Oxford Nanopore Technology (ONT) launched the first commercially available high-throughput nanopore-based sequencing platform, which provides significantly longer reads than existing technologies[15]. In this report, "nanopore sequencing" or "nanopore sequencer" refers to the sequencing service or sequencer provided by ONT, although there may be other nanopore-based technologies.

1.3.1 Nanopore sequencing in transcriptomics

Table 1.1 shows the main advantages and disadvantages of nanopore sequencing compared to Illumina. Nanopore long reads can cover the full length of mRNAs, which will significantly benefit transcriptome analysis, especially for identifying exon-to-exon connections with higher sensitivity[16] (Figure 1.1B). In addition, unlike the short-read sequencing, nanopore sequencing does not require the amplification of cDNA. It also allows direct sequencing of native RNA, which avoids the bias introduced by the reverse transcription and PCR amplification process, and potentially offers better quantification accuracy[17].

However, nanopore reads have a considerably higher error rate, and the read depth is normally lower, due to the relatively low throughput, making the analysis of nanopore reads challenging. New bioinformatics tools that can deal with the high error rate and accurately identify and quantify transcripts will be desirable ~~in the field~~.

Table 1.1: Characteristics of selected sequencing platforms

Platform	Maximum read length	Throughput per run	Error type	Read accuracy	Machine cost	Cost per Gb of DNA
Illumina Miseq [18, 19]	~300 bp	15 Gb	mismatch	~99.9%	$128,000	$500
Illumina Hiseq [18, 19, 20]	~150bp	150 Gb	mismatch	~99.9%	$750,000	$40
Oxford Nanopore MinION Mk1B [21, 22, 23]	~2 Mbp*	20 Gb	Indel/mismatch	~90%	$1,000	$15-60

* Theoretically, there is no limit for maximum read length for nanopore technology itself. The difficulties come from the extraction of extremely long DNA in the library preparation step.

1.3.2 How does Nanopore sequencing work?

This section provides a summary of basic nanopore sequencing principle[24, 25]. To sequence DNA or RNA molecules, a membrane with nanoscopic protein pores is set under electric current (Figure 1.2A). The current flow can only go through the pores since the membrane is electrically resistant. The molecules transit through the pores from one side of the membrane to the other, and the nucleotide bases inside the pores cause changes in the electric current depending on the combination of bases in the pore (usually referred as a k-mer, namely the sequence consists of k bases). Figure 1.2B shows the different expected current levels that are caused by different k-mers. The raw output from nanopore sequencer is a current trace and is also called a "squiggle". The squiggle is then translated into nucleotide sequence using computational tools (Figure 1.2C). The translation process is usually referred to as "basecalling" and the computational tool(s) is called a "basecaller". The basecalled sequences will be referred to as "nanopore reads" in this report. In the earlier pore version (R7, 7.3), the current level at the moment was mainly dependent on a k-mer of length 5 or 6 occupying the pore. In the more recent

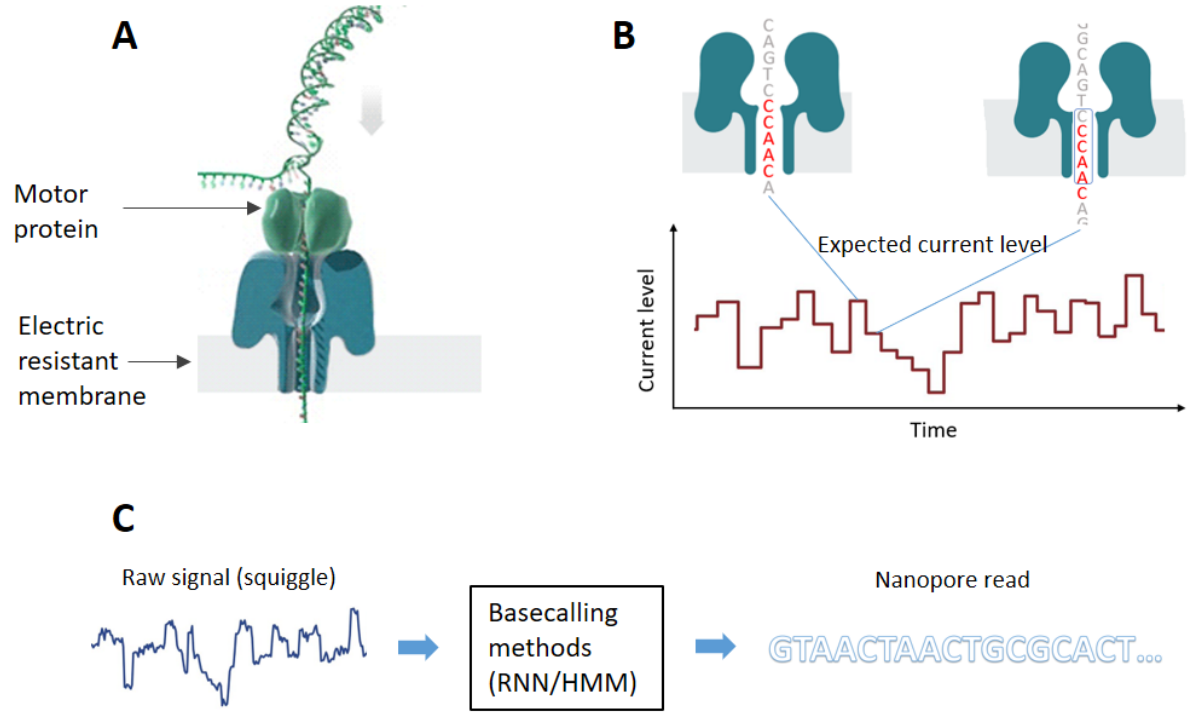

Figure 1.2: The nanopore sequencing technology
Figure A and B were modified from [25]. A: The principle of the sequencing process: a single stranded molecule transits through a pore on the membrane under an electric current. A sensor detects the change of the current level inside the pore. B: Explanation of the expected signal: Different expected values depend on the k-mer inside the pore at that time. C: Basecalling: the raw signal (squiggle) will be translated into a nucleotide sequence (nanopore read) using a recursive neural network (RNN) or Hidden Markov Model (HMM).

pores (R9.4), it is reported that the 3-mer in the centre of the pore mainly determines the current level, with less effect from the other bases inside the pore[26]. To have enough signal for reliable k-mer identification, ONT attaches a motor protein (Figure 1.2A) to the nucleic acids to be sequenced to slow down the translocation speed[26, 27, 28, 29].

1.3.3 Errors in nanopore sequencing data

As mentioned above, nanopore sequencing data has a relatively high error rate. There are potentially two sources of errors. The first is that the squiggle itself is too noisy and the low signal-to-noise ratio makes it too challenging to identify the nucleotide sequence. The second is that the error occurs in basecalling, which translates the squiggle into nanopore

reads using computational tools. There are several factors that make the interpretation of squiggles difficult. Even with the motor protein slowing the speed of translocation of the DNA or RNA molecules, the translocation speed is still nonuniform so that a different number of data points will be generated across k-mers since the current level sensor measures at a constant frequency[28, 30]. Deletion/insertion errors tend to arise often when the dwell time (the time of a k-mer occupies a pore) is too short/long. It's also challenging to distinguish the adjacent k-mers which produce similar current levels. A good example is homopolymer regions, in which the adjacent k-mers are identical in sequence so that a lot of deletion/insertion errors will happen in these regions.

1.4 Limitation of nanopore reads in identification of splice sites

In section 1.3.1, I mentioned the advantages of using nanopore reads to study alternative splicing. Nanopore reads are good at identifying which exons are present in a transcript. However, due to the high error rate, nanopore reads have limited abilities to accurately identify the splice sites.

The main type of alternative splicing is exon skipping, in which a whole exon is included or excluded in different isoforms (accounting for 38% of conserved alternative splicing events between human and mouse[31]). Additionally, alternative splicing can also alter which bases form the splice site, which is called alternative 5' splice sites (A5ss) and alternative 3' splice sites (A3ss) (Figure 1.3A). A5ss and A3ss account for ∼18% and ∼8% in human-mouse conserved splicing events respectively, but are relatively poorly characterised[31]. The presence of A5ss and A3ss make the mapping results of Nanopore reads hard to interpret. For instance, Figure 1.3B shows a typical splice site mapping result where the mapped reads support two different splice sites on the 5' side. This is confusing because it could be caused by sequencing or mapping error, or both of the supported splice sites may be true A5ss.

Some computational tools provide splice site mapping correction methods, which are usually applied to identify splice sites for nanopore reads. For example, pipelines such as pinfish[32] perform the correction by assuming the reads with similar exon/intron structure share the same splice sites, and choose the splice sites supported by most reads. Some other methods, such as FLAIR[33], use annotated splice sites or the sites supported by the short-read data. Both approaches have limitations. Methods like pinfish are not aware of the location of known A3ss and A5ss. For methods that use splice site

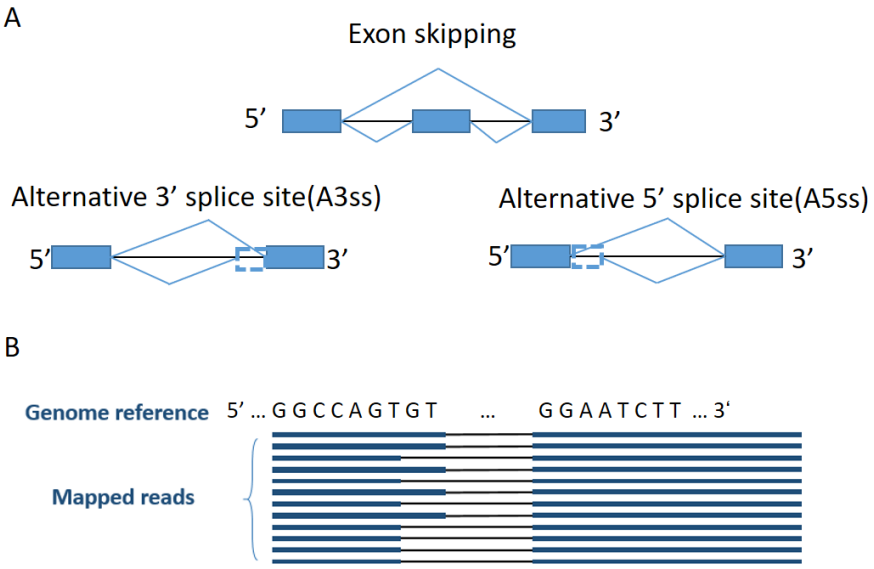

Figure 1.3:
A: Illustration of exon skipping, alternative 3' splicing site(A3ss), alternative 5' splicing (A5ss). The blue boxes represent the exons and the black lines in between is the introns. The blue lines connecting the exons represent the alternative splicing. B: Example of fuzzy exon boundary observed in nanopore reads mapping.

annotations, it is hard to find novel splice sites and for most organisms annotations are poor, (even for humans they are still far from complete). Generating short-read sequencing as extra information is helpful, but it increases the costs. Furthermore, in some circumstance, short-reads covering whole transcripts will not be available. For example, single-cell RNA sequencing technology has great potential of studying the gene expression in different cell types, but the most commonly used protocol is 10X scRNA-seq which can only sequence the 3' end of a transcript. Hence methods that can accurately identify splice sites using nanopore sequencing data only are required.

1.5 Aim of the project

All the current bioinformatics tools for nanopore that perform splice site correction use the nanopore reads. However, nanopore data is not limited to the nanopore reads. Relative to the nanopore reads, the squiggles should have more information, since the basecalling process is error-prone. Previous studies [34, 35] have shown that squiggles can be successfully used for sequence motif identification with high accuracy.

In this project, I am going to develop a method that can accurately identify and quantify splice sites using squiggles. It will help study mRNA isoforms in higher resolution with nanopore sequencing.

Methods

2.1 Overview of NanoSplicer workflow

This section provides a brief overview of "NanoSplicer" (Figure 2.1), which is a method I developed for better identification of splice sites. Details of the workflow will be introduced in the following sections. Input data to NanoSplicer are nanopore reads, their squiggles, and a reference genome sequence. First, I map nanopore reads to the reference genome sequence and locate exon junctions supported by the reads (Figure 2.1A). The mapped reads tend to show unclear splice sites. Thus, for a given exon junction, I perform the following steps to improve the identification of splice sites. NanoSplicer chooses candidate splice sites and obtains an expected squiggle for each candidate (Figure 2.1B). The expected squiggles will be referred to as "candidate squiggles" in this report. For each read supporting the exon junction, I extract a part of its squiggle corresponding to the location of the exon junction (Figure 2.1C). The extracted squiggles will be referred to as "junction squiggles" in this report. Finally, the NanoSplicer model outputs the probabilities of assigning each junction squiggle to candidates squiggles using similarities between junction squiggle and candidate squiggles.

2.2 Basecalling and mapping nanopore reads

Squiggles are basecalled into nanopore reads using basecalling methods. For example, following the typical Oxford Nanopore sequencing pipeline, I use the latest version of Guppy basecaller (https://community.nanoporetech.com/downloads/guppy) for my analysis in this report.

Input data to the NanoSplicer workflow are the squiggles, nanopore reads, and a reference

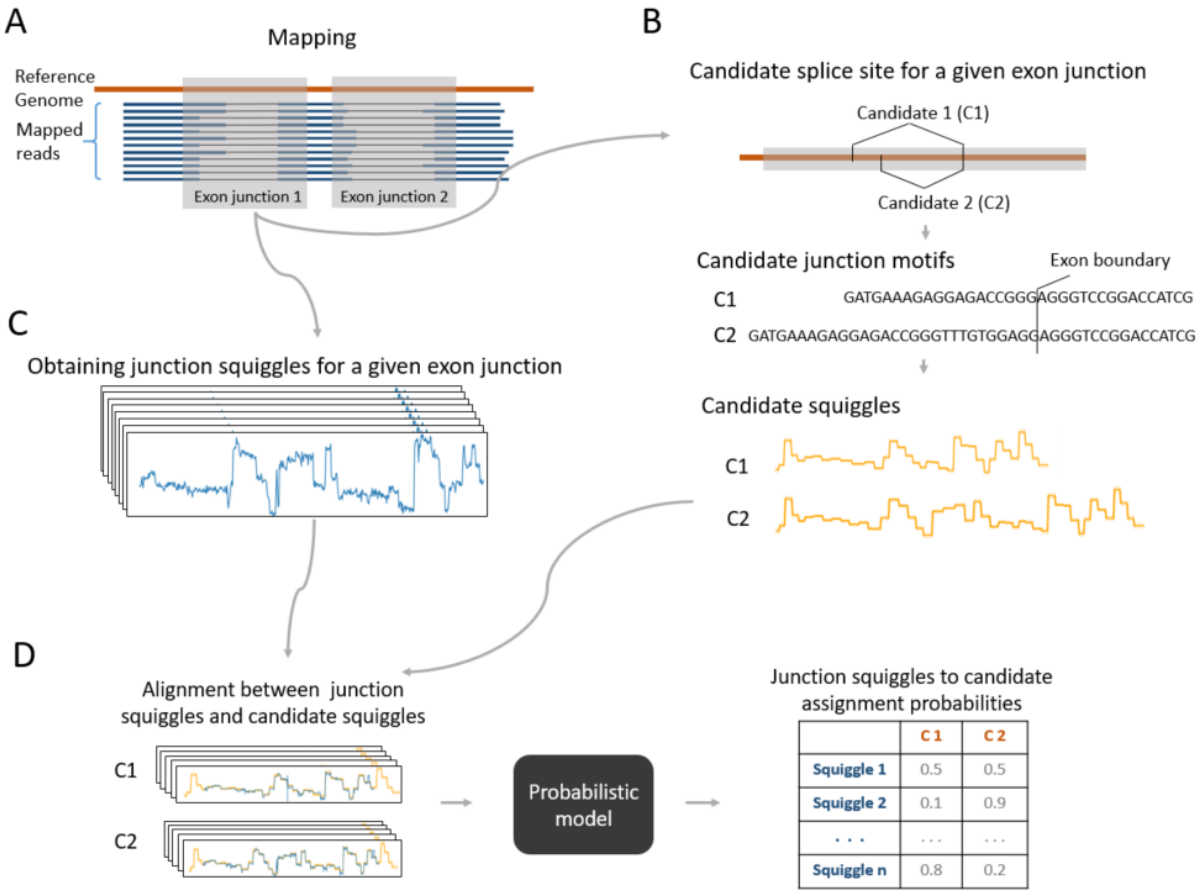

Figure 2.1: NanoSplicer workflow
A. Map nanopore reads to a reference genome: mapped reads support two exon junctions in this example. Currently, NanoSplicer performs splicing site identification for each of the exon junctions independently. B: A set of candidate splicing sites are a required input. Candidates can be specified by the user or by a procedure implemented in NanoSplicer (details in the section 2.3.1). The splice site will be converted to junction motifs and then the candidate squiggles in NanoSplicer. C: Find the specific part of the squiggle that comes from a given exon junction (grey box in A). D: Each junction squiggle will be aligned to each candidate squiggle, and a probabilistic model will be applied to output the assignment probabilities from junction squiggles to candidate squiggles.

genome sequence. As a first step, I map the reads to the reference genome sequence using minimap2[36] (https://github.com/lh3/minimap2), and locate exon junctions supported by the mapped reads. Minimap2 has been developed for mapping of long reads and it also allows splice-aware mapping. As mentioned in section 1.3.1, the long-read mapping results are relatively good at locating the exons, and also at identifying the exon-to-exon connections. The rough exon location in the mapping result is robust, especially for exons matching annotated exons. However, the mapping results tend to show unclear exon boundaries (or splice sites). Therefore, NanoSplicer performs the following steps for each exon junction to improve the identification of splice sites.

2.3 Construction of candidate squiggles

For each exon junction, NanoSplicer constructs candidate squiggles by first choosing candidate splice sites, and then obtaining an expected squiggle for each candidate.

2.3.1 Candidate splice sites

As mentioned above, it is often observed that the exon boundary mapping of nanopore reads will be fuzzy inside a small window. NanoSplicer allows users to provide their candidate splice sites according to their specific interest. There are a few options for users to do this, such as using annotations or splice sites supported by short reads. Otherwise, NanoSplicer generates the candidate splice sites as follows. It has been reported that ∼98.7% of splice sites are canonical, which means the introns start with GT and end with AG[37]. Currently, NanoSplicer considers all the splice sites following the GT-AG pattern within a window around the exon junction. NanoSplicer uses 20 nt as the window size by default, but it also allows users to specify it. In addition, NanoSplicer considers splice sites supported by at least 3 mapped reads. Users can also specify the threshold for the read count. NanoSplicer can also include annotated splice sites as the candidates when available.

2.3.2 Obtaining "candidate squiggles": expected squiggle for each candidate splice site

Given the candidate splice sites for the exon junction, I first form an exon junction motif for each candidate splice site by connecting two sequences from both sides of the

exon boundary (Figure 2.1B). In this step, NanoSplicer doesn't take genetic variants into account, and all the bases in the junction motifs will come from the reference genome instead of the nanopore reads. To ensure the differences between the candidate junction motifs is only caused by the difference in the splice sites, the junction motifs have different lengths to share the same start and end bases, and the differences will only be at the centre.

After obtaining the candidate junction motifs, NanoSplicer generates candidate squiggles using Scrappie (https://github.com/nanoporetech/scrappie). Scrappie takes a sequence motif as input, and uses a squiggle prediction model to return an expected squiggle that contains mean, variance and dwell time for each base (Figure 2.1B). The current level mainly depends on the nucleotides in the centre of the pore, but it also depends on a couple of nucleotides before and after the central bases. Thus, the expected quiggle from Scrappie for the start and end bases in the input motif will be less reliable. For example, if the Scrappie takes a sequence motif "AGGCAGCTGA" as input, the expected current level for the first "A" will be less accurate since it also depends on the bases before the first "A". A similar phenomenon affects the bases at the end. To avoid this, NanoSplicer uses a longer sequence of bases as input and then only takes the result from the bases of interest in the middle.

2.4 Obtaining junction squiggles

For each read supporting the exon junction, I locate the subset of its squiggle belonging to the exon junction (Figure 2.1C) using Tombo (https://github.com/nanoporetech/tombo). The subset of the squiggle is referred to as "junction squiggle". Tombo provides a tool called "resquiggle" which allows for mapping of the squiggle to a nucleotide sequence, that is, assigning data points in the squiggle to each base on the sequence. NanoSplicer maps the squiggle to the reference genome using the resquiggle tool and extracts the junction squiggle (Figure 2.2). The resquiggle tool does not support spliced mapping, which means the introns should not be included in the the sequence for Tombo input. NanoSplicer inputs a transcript reference to Tombo. In some cases the transcript reference is slightly different from the true sequence, for instance due to the genetic variants. The squiggle-to-sequence mapping will then be inaccurate. NanoSplicer handles this issue at the alignment step in the section 2.5.2.

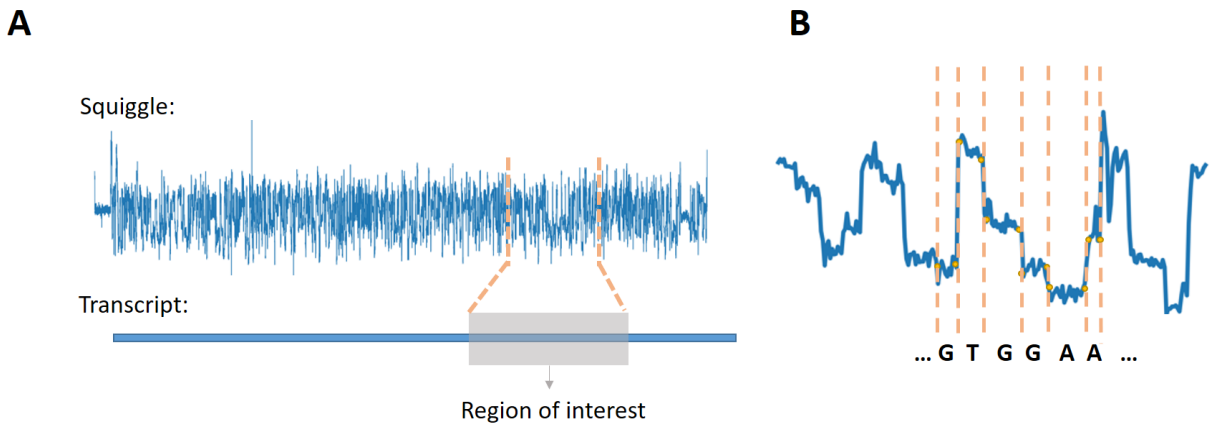

Figure 2.2:
A: Locating the subset of squiggle that matches to a specific region in a sequence using Tombo resquiggle (https://github.com/nanoporetech/tombo) B: A closer look at Tombo output. Tombo assigns the data points in the squiggle to each base in a given sequence.

2.5 Identification of splice sites

Now, for each exon junction, NanoSplicer has constructed the candidate squiggles (Section 2.3) and extracted the junction squiggles (Section 2.4). Before identifying splice sites using the candidate and junction squiggles (Section 2.5.3), NanoSplicer will normalise the squiggles so that they are comparable (Section 2.5.1). In addition, to take into account the non-uniform translocation speed of the molecules, NanoSplicer will align the squiggles so that their time axes are comparable (Section 2.5.2).

2.5.1 Normalisation of squiggles and removal of potential outliers

Let $y = (y_1, \ldots, y_B)$ represent a squiggle with a length B, where y_b denotes the current level at time b. Following other squiggle processing tools [38, 39], NanoSplicer normalises the squiggle y by using

$$\tilde{y}_b = \frac{y_b - \text{median}(y)}{\text{MAD}} \quad \text{for} \quad b = 1, \ldots, B, \tag{2.1}$$

where MAD denotes the median absolute deviation defined by

$$\text{MAD} = \text{median}(|y_1 - \text{median}(y)|, \ldots, |y_B - \text{median}(y)|). \tag{2.2}$$

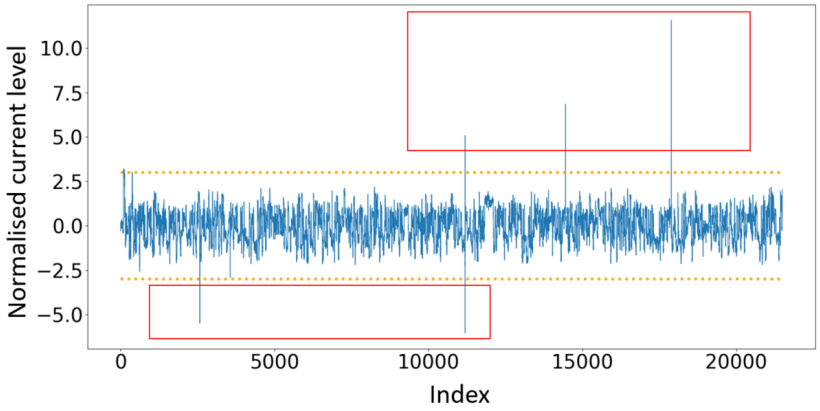

Figure 2.3: An example of clear current spike within a squiggle
Transient current spikes are usually observed in a squiggle (red box), NanoSplicer removes all points with absolute values larger than a user-specified threshold. Orange line is the default threshold, which is 3 in NanoSplicer.

The median and MAD are more robust to outliers than mean and standard deviation, so they are more suitable for squiggle data which sometimes show transient current spikes. Figure 2.3 shows an example of squiggle with multiple spikes. NanoSplicer removes the spikes if the absolute value of the normalised squiggle is bigger than a threshold. The length of the squiggle is reduced after the spike removal. Users can specify the threshold for a spike. NanoSplicer uses 3 as a defaut threshold since the normalised value in squiggle is mostly observed to lie between -2.5 and 2.5.

In the remainder of section 2, I assume that junction and candidate squiggles are normalised as described in this section.

2.5.2 Aligning junction squiggle to each of candidate squiggles

Dynamic Time Warping (DTW): NanoSplicer uses Dynamic Time Warping (DTW) [40] for aligning junction and candidate squiggles and measuring their distance or similarity. Time-rigid measurement (Figure 2.4A) of distance is less preferred since there could be temporal distortions in time axis. DTW is an efficient algorithm for aligning and measuring the distance between two sequences which may vary in speed (Figure 2.4 B). This section provides a brief intuitive description of DTW (Figure 2.4 B and C). Further, more formal, background on DTW can be found in [41]. Suppose I have two sequences, $Q = (q_1, \ldots, q_n)$ and $C = (c_1, \ldots, c_l)$, with varied speed. DTW considers an $n \times l$ matrix M, where n and l are lengths of Q and C, respectively, and the (i, j)-th element of the matrix, M_{ij}, contains a distance between q_i and c_j (Figure 2.4B). Every possible warping alignment between Q and C can be presented as a path through M from M_{11} to M_{nl}. A

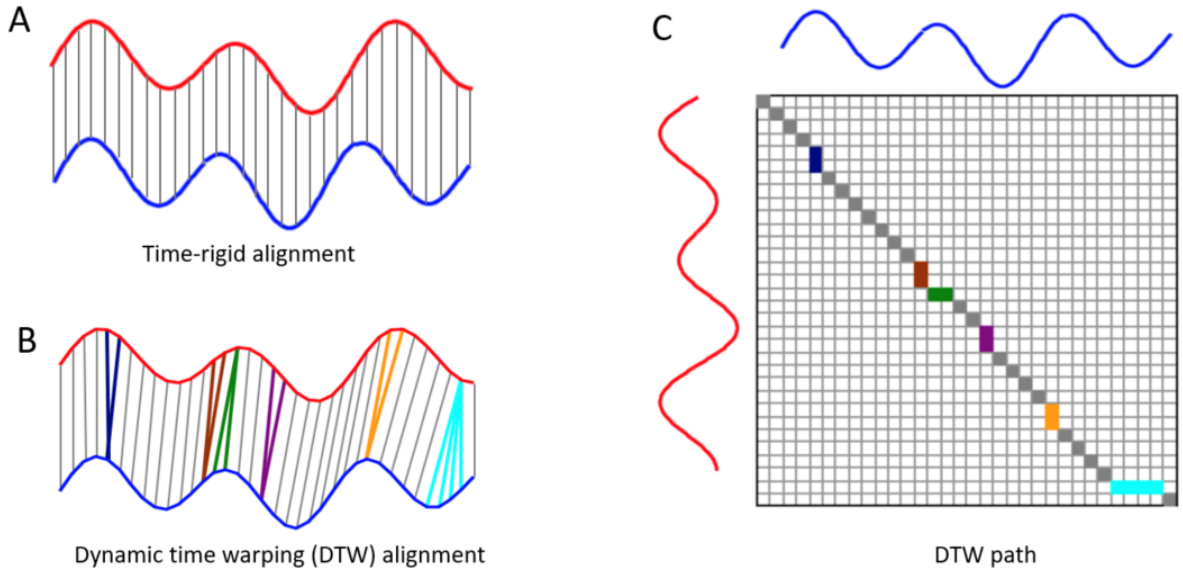

Figure 2.4: Explanation of dynamic time warping (DTW)
This figure was modified from [41] A: A time-rigid alignment between two sequences (blue and red). Black line connects alignment between two points in two sequences respectively. B: Dynamic time warping (DTW) alignment between the same sequences in A, the colours of the lines match to the colours in Figure C. C: A path in a distance matrix that presents the alignment in B.

path passing M_{ij} indicates that there is an alignment between q_i and c_j (Figure 2.4 B and C). DTW finds the path p which minimises the distance (defined as $\sum_{(i,j)\in p} M_{ij}$) between the sequences by using dynamic programming.

Aligning candidate and junction squiggles using DTW: NanoSplicer treats junction squiggles (Figure 2.1C) as observations sampled from one of candidate squiggles, which contain means and standard deviations (Figure 2.1B), and uses a negative log likelihood as the distance in DTW. Specifically, let $X = (X_1, \ldots, X_n)$ be a junction squiggle with length n and let $\mu = (\mu_1, \ldots, \mu_l)$ and $\sigma = (\sigma_1, \ldots, \sigma_l)$ be the means and standard deviations for a candidate squiggle with length l. Then, the distance between X_i and (μ_j, σ_j), the (i, j)-th element of the DTW matrix, is defined by

$$M_{ij} = -\log f(\frac{|X_i - \mu_j|}{\sigma_j}), \tag{2.3}$$

where f is the probability density function of the standard normal distribution. Here, I didn't use the normal density function with mean μ_j and standard deviation σ_j. Because σ_j normally a small value, the negative log density could be a negative value when X_i is close to μ_j, which makes it an inappropriate distance function. DTW with the negative log likelihood distance will provide the path with the minimum distance and the corresponding

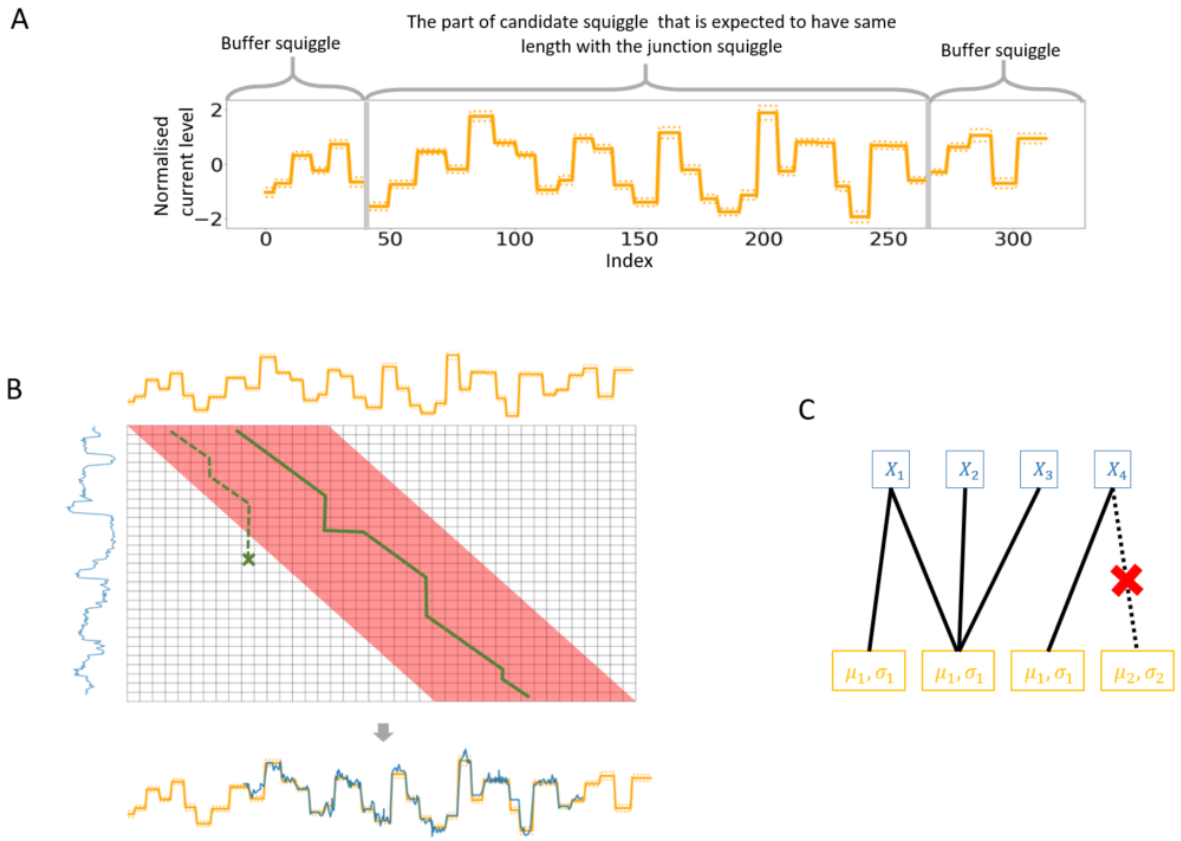

Figure 2.5: Implementation of DTW in NanoSplicer
A: The constitution of candidate squiggle in NanoSplicer. The candidate squiggle is presented as a sequence of means (solid line) and standard deviations (the dashed lines indicate where are the mean +/- 1 standard deviation). The junction squiggle is expected to match the middle part (between grey lines) of the candidate squiggle if the junction squiggle extraction is accurate. NanoSplicer includes some buffer squiggle at both end to allow for small shifts in the junction squiggle boundaries. B: The start and end of the DTW alignment are allowed to be in the middle of the candidate squiggle. A band (in red) is applied to restrict the possible paths and the green lines are examples of a valid path (solid line) and a invalid path(dashed line). C: Each point in the junction squiggle (blue) can only be aligned to a unique mean and standard deviation. Black lines present the valid (solid line) and invalid (dashed line) alignment given the alignment of previous points.

warping alignment.

As mentioned in section 2.4, there is no guarantee that the junction squiggle has the same start and end positions as the candidate squiggle. To account for this, NanoSplicer includes more bases in the candidate squiggles as a buffer, so the candidate squiggle is longer than the junction squiggle to ensure the candidate squiggle contains the expected shape of the junction squiggle. Figure 2.5A shows an example of how a candidate squiggle is formed. Hence, a global alignment between the two squiggles are not required. In the implementation of DTW, I allow DTW alignment to start and end in windows near the start and end of the candidate squiggle (Figure 2.5B). Equivalently, the best DTW path is allowed to start from somewhere in the first row that is near M_{11} instead of strictly M_{11}. A similar idea is applied to the end of the path.

In addition, there are some restrictions for valid DTW paths in NanoSplicer. To avoid the time axis being warped too much, NanoSplicer limits the valid DTW path to stay inside a diagonal band with user-specified bandwidth (Figure 2.5B). Furthermore, NanoSplicer doesn't allow data points in a junction squiggle to be aligned to different means/standard deviations in a candidate squiggle. For example, in Figuer 2.5C, the X_4 can not be aligned with both μ_1, σ_1 and μ_2, σ_2. Because it is not interpretable that there are two different expected values for an observed one.

After this section, each data point in the junction squiggle is aligned to a unique mean and standard deviation, which enables the likelihood calculation in section 2.5.3.

2.5.3 NanoSplicer model: accurate identification of splice sites

I build a mixture model to accurately identify splice sites among the candidates. Suppose I have K junction squiggles, $(X^k)_{k=1}^K$, and M candidate squiggles, $C = (C^m)_{m=1}^M$, for a given exon junction. Let Z^k denote the (unknown) candidate site of origin of the k-th junction squiggle. Assuming independence across junction squiggles, the probability of the K junction squiggles given the M candidate squiggles can be written as $\Pr((X^k)_{k=1}^K|C) = \prod_{k=1}^K \Pr(X^k|C)$. In the following section, I first detail our model for one junction squiggle (I drop the superscript ^{k} for simplicity).

Mixture model: A mixture model for a given junction squiggle X can be written as

$$\Pr(X|C) = \sum_{m=1}^{M} \Pr(X|Z = m, C) \Pr(Z = m|C). \tag{2.4}$$

Assume that I have no information about the candidate splice site of origin of each junction squiggle and that the probability that the junction squiggle coming from splice site m is the same for all m,

$$\Pr(Z = m|C) = \frac{1}{M}. \tag{2.5}$$

Given the alignment between the junction squiggle and the m-th candidate squiggle, I compute $\Pr(X|Z = m, C)$ as follows. Let $X = (X_1, \ldots, X_n)$ represent the junction squiggle with length n and let $C^m = (\mu_1^m, \ldots, \mu_l^m, \sigma_1^m, \ldots, \sigma_l^m)$ be the means and standard deviations for the m-th candidate squiggle with length l. I assume that the current levels are independent conditional on their means and standard deviations and that they follow normal distributions, leading to

$$\Pr(X|Z = m, C) = \prod_{i=1}^{n} \Pr(X_i|C^m) = \prod_{i=1}^{n} f(X_i|\mu_j, \sigma_j), \tag{2.6}$$

where μ_j and σ_j are the mean and standard deviation that are aligned to X_i in the alignment from the modified DTW (Section 2.5.2), and f is the probability density function of the normal distribution with a mean of μ_j and a standard deviation of σ_j.

Assignment probability of each junction squiggle: The assignment probability of the k-th junction squiggle to the m-th candidate splice site can be computed by

$$\Pr(Z^k = m|X^k, C) = \frac{\Pr(X^k|Z^k = m, C)\Pr(Z^k = m|C)}{\sum_{m'=1}^{M} \Pr(X^k|Z^k = m', C)\Pr(Z^k = m'|C)} \tag{2.7}$$

$$= \frac{\prod_{i=1}^{n} \Pr(X_i^k|C^m)}{\sum_{m'=1}^{M} \left[\prod_{i=1}^{n} \Pr(X_i^k|C^{m'})\right]}. \tag{2.8}$$

2.5.4 NanoSplicer certainty : quantifying certainty on whether the candidates contain the true splice site for each junction squiggle

The NanoSplicer mixture model assumes that the candidates contain the true splice sites for the exon junction, but this assumption will not always hold in practice. Also, in our data analysis (section 3.2), I observed that Tombo sometimes fails to locate correct junction squiggles corresponding to the exon junction. In either cases, none of the candidate squiggles contains the expected shape of observed junction squiggle, so the assignment probabilities from NanoSplicer are less reliable. Therefore, I proposed a procedure to

quantify certainty on whether the candidates squiggles contain the expected shape for each junction squiggle. This certainty can be potentially an indicator of unreliable assignment from junction squiggles to candidate squiggles.

I use the largest similarity between a given junction squiggle and the candidate squiggles to quantify the uncertainty because it is more likely to have larger (smaller) value when the candidates (do not) contain the true splice site for junction squiggles. In NanoSplicer, the similarity between the k-th junction squiggle X^k (with length l^k) and a candidate squiggle C^m is defined as

$$S_m^k = \frac{\sum_{i=1}^{l^k} logP(X_i^k | Z^k = m, C)}{l^k} \tag{2.9}$$

In formula 2.9, the log-likelihood is adjusted by l^k to ensure S_m^k for different k are in a similar scale. Suppose S^k represent the largest similarity among all candidates for the X^k:

$$S^k = \max_{m \in 1,\ldots,M} S_m^k. \tag{2.10}$$

Let H^k denote a binary indicator for whether the M candidates contain the expected shape for the k-th junction squiggle ($H^k = 1$). I assume that $S^1, \ldots, S^k$ are independent, and the probability of S^k is given by

$$\Pr(S^k) = \Pr(S^k | H^k = 1) \Pr(H^k = 1) + \Pr(S^k | H^k = 0) \Pr(H^k = 0). \tag{2.11}$$

NanoSplicer approximates $\Pr(S^k | H^k = 1)$ and $\Pr(S^k | H^k = 1)$ using empirical distributions. Details on how to obtain the empirical distributions will be described in the section 3.2. Users can specify values for $\Pr(H^k = 1)$, or they can be learned from data using empirical Bayes methods.

Quantifying NanoSplicer certainty: The certainty on whether the candidates contain the true splice site for the k-th junction squiggle can be quantified by

$$\Pr(H^k = 1 | S^k) = \frac{\Pr(S^k | H^k = 1) \Pr(H^k = 1)}{\Pr(S^k | H^k = 1) \Pr(H^k = 1) + \Pr(S^k | H^k = 0) \Pr(H^k = 0)}. \tag{2.12}$$

cDNA sequin data analysis

Sequins[42] are synthetic spliced mRNA transcripts whose splice sites, sequence, and quantity are known. To assess the performance of NanoSplicer, I analysed nanopore data from sequins where true splice sites are given. Dr. Mike Clark generated a nanopore cDNA dataset with 4 technical replicates of a differentiated SHSY-5Y sample containing Sequin v2 controls. Libraries were made from a pool of polyA+ RNA using the equivalent of 1ug total RNA starting material. Library preparation utilised the Oxford Nanopore cDNA-PCR kit (PCS108) with LWB001 barcodes 7-10. Squiggles were basecalled using Guppy v3.0.3.

To identify reads from sequins, I collected the reads that were successfully mapped to the synthetic sequin reference genome using minimap2 v2.17. Sequin are designed to be different in sequence from any biological DNA or RNA, so the mapped reads can be considered as confidently from the sequins. I obtained 3570 reads mapped to 32 transcripts over 23 sequin genes, leading to 11585 junctions within reads (Figure 3.1) over 144 exon junctions. Among 11585 junctions within reads, 9782 (84.4%) were mapped to true splice sites.

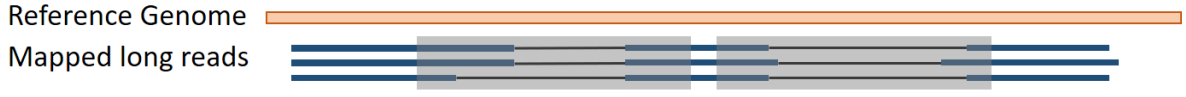

Figure 3.1: Definition of "junctions within reads":
The subsets within reads support exon junctions (grey box). This figure shows 6 junctions within reads.

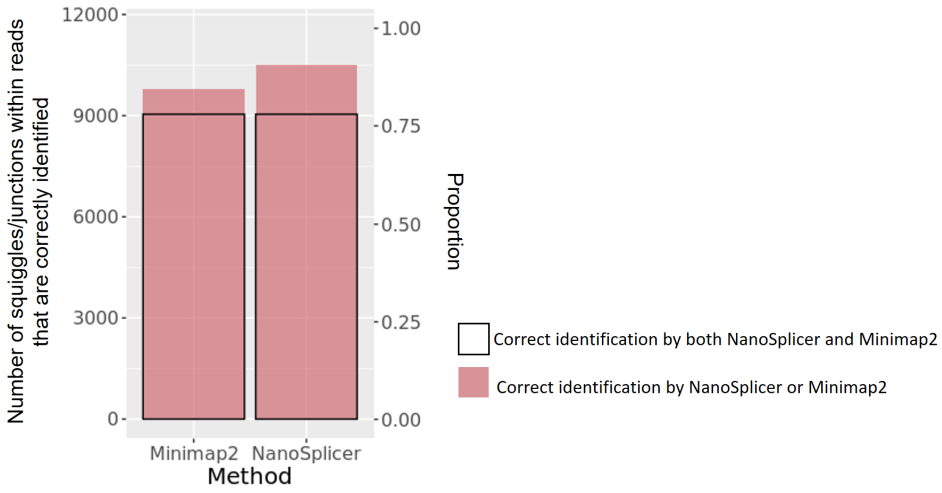

Figure 3.2: Comparison between NanoSplicer and minimap2
Number and proportion of junction squiggle/junctions within reads whose splice site are correctly identified by NanoSplicer and minimap2.

3.1 NanoSplicer improves upon the original mapping results

To improve the identification of splice sites by the minimap2 mapping, I applied NanoSplicer to the 3570 nanopore reads, their squiggles, and the sequin reference genome. For a given exon junction, I constructed candidate splice sites that contain the true splice site and canonical splice sites within a 20-base window centred at the true one. In DTW, the bandwidth was set to 40% of a candidate squiggle length.

NanoSplicer computed assignment probabilities for each of the 11585 junction squiggles. NanoSplicer correctly assigned 10515 (90.8%) junction squiggles to the true splice sites with the highest probability, leading to an increase of 6.4% in the number of correct identifications compared to the minimap2 mapping results (Figure 3.2). Among 1803 junctions in reads that have been mapped to wrong splice sites by minimap2, 1473 (81.7%) were successfully corrected by NanoSplicer. Figure 3.3 shows an example of a splice site mapping corrected by NanoSplicer. The red read in Figure 3.3A was mapped to a wrong splice site due to a 6-base deletion during basecalling. Figure 3.3B shows that the junction squiggle from the read is more similar to the candidate squiggle for the true splice site than that for the wrong one, which allows NanoSplicer to identify the true splice site. This illustrates that NanoSplicer effectively uses squiggle similarities to better identify splice sites that were missed by the original mapping due to errors in the basecalling.

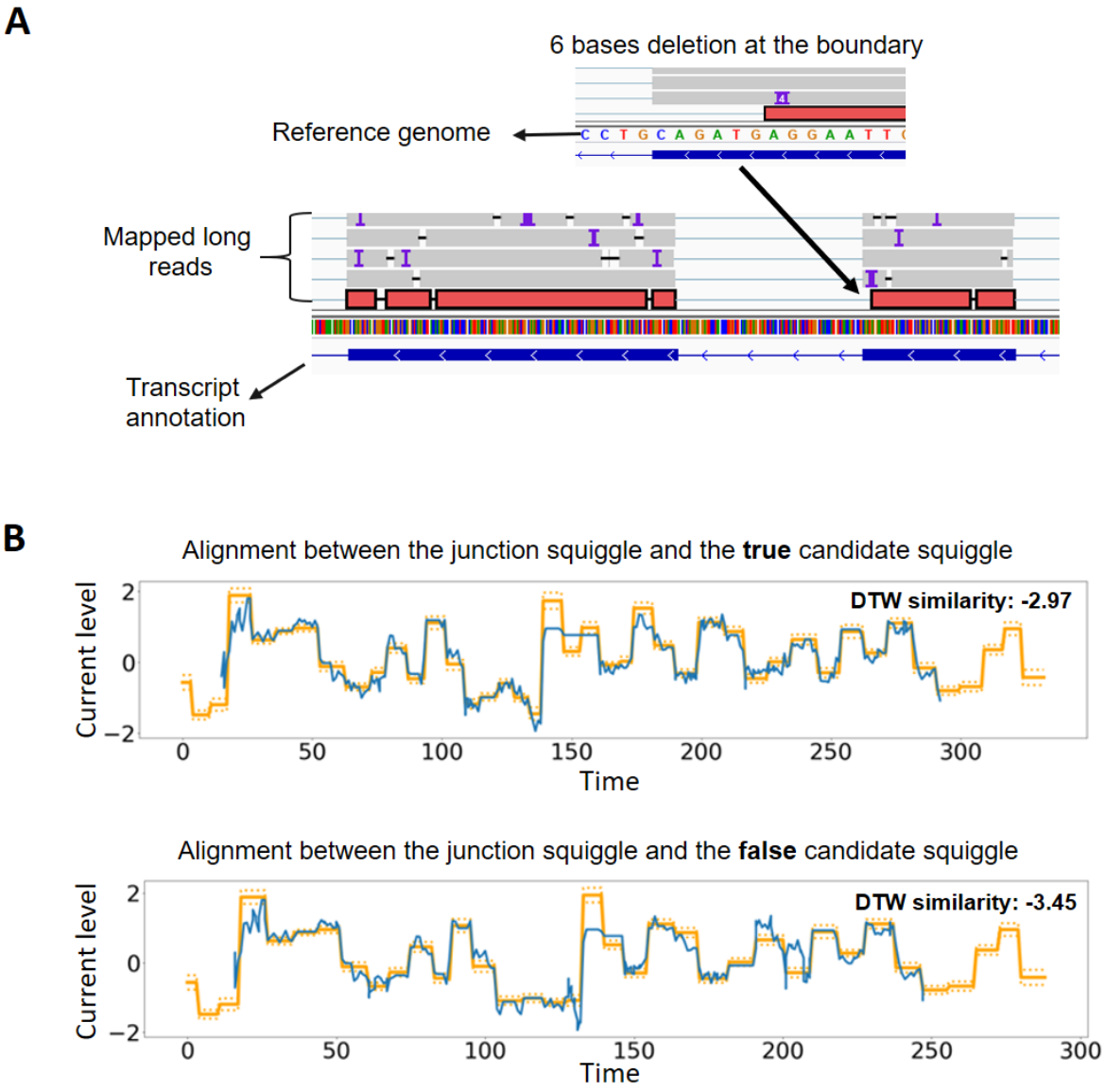

Figure 3.3: An example of splice site mapping corrected by NanoSplicer

A: An example Sequin read (red line) at an exon junction. A 6-based shift was observed on the donor site. The true site is provided in the transcript annotation (blue line). Incorrect mapping was caused by basecalling errors. B: DTW visualisation of the red read from A. The candidate squiggles (yellow) are aligned to the junction squiggle (blue). The candidate squiggle in the top panel is from the annotated (true) splicing site. The candidate squiggle in the bottom panel is from the mapping result with 6-base shift. The true candidate squiggle has a higher similarity (defined in section 2.5.4) to the junction squiggle, which means NanoSplicer will assign this junction squiggle to the correct candidate.

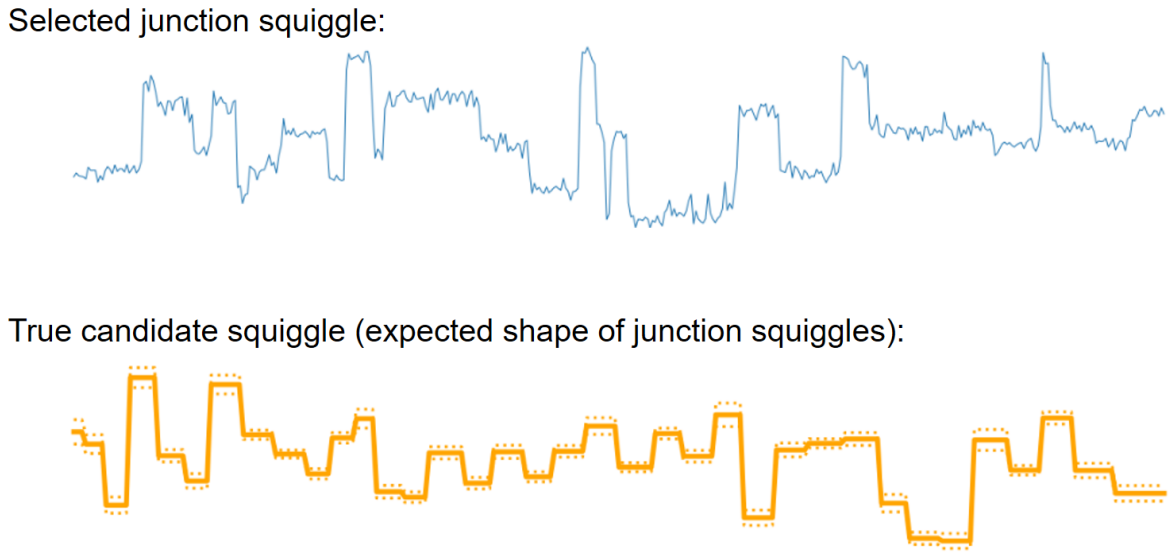

Figure 3.4: Example of potential error in Tombo output.
The selected junction squiggle looks very different with the expected shape of junction squiggles.

3.2 NanoSplicer certainty

The analyses in section 3.1 assumed that the candidates contain the true splice sites for the exon junction, but this assumption will not always hold in practice. Also, as shown in Figure 3.2, 7.6% of the time a junction within a read was mapped correctly NanoSplicer selected the wrong splice site. It is undesirable that NanoSplicer introduced splice junction calling errors in some cases. By inspecting these cases, I found that sometimes the selected junction squiggle looks very different from its expected shape (Figure 3.4). A possible explanation is that the location of junction squiggle obtained from Tombo is sometimes inaccurate. The selected "junction squiggle" doesn't actually come from the part of the read containing the junction to be tested but from another location. This observation motivated me to measure the (un)certainty of NanoSplicer using the model described in section 2.5.4.

As described in section 2.5.4, $Pr(S^k|H^k)$ is approximated from empirical distributions. For the sequin reads, the true splice site for each junction squiggle is known, making it possible to control whether the expected shape (true candidate squiggle) of a junction squiggle is either included ($H^k = 1$) or excluded ($H^k = 0$) in the candidates. I split the 11585 junction squiggles into two groups (H1 and H0) with the same size. In group H1, I included the true candidate in the candidate set for each junction squiggle. In group H0, I used a candidate set with the true candidate removed for each junction

squiggle. Given the candidate set for the junction squiggle X^k, I calculated S^k using formula 2.9 and 2.10. The empirical distribution of $S^k|H^k = 1$ and $S^k|H^k = 0$ can be obtained by density estimation of S^k in group H1 and H0 respectively (Figure 3.5A). Although, there could still be some cases in group H1 whose expected shapes are not included in the candidate set ($H^k = 0$), when the selected junction squiggles do not come from the junctions within reads due to the inaccurate output from Tombo. Based on the result from previous sections, NanoSplicer gave correct splite site identification for most of the junction squiggles. It was reasonable to believe that most of the times the candidate squiggles contain the expected shape of junction squiggles in group H1, so that the empirical distribution should not be affected too much.

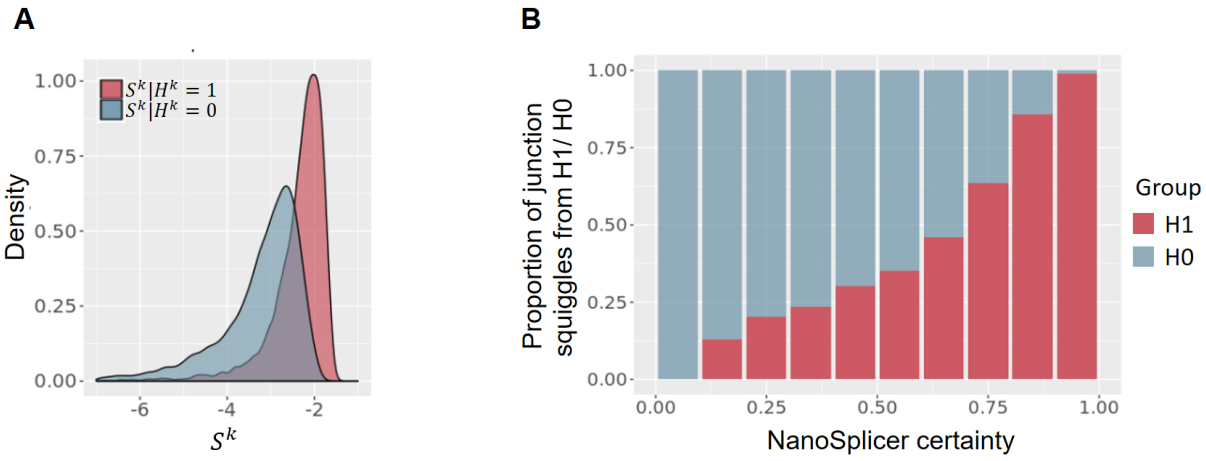

Figure 3.5:
A: Empirical distributions of $S^k|H^k = 1$ and $S^k|H^k = 0$. B: Each bar shows the proportion of junction squiggles with specific NanoSplicer certainties that are from H1 or H0.

3.2.1 Simulation shows that "NanoSplicer certainty" quantifies potential uncertainty introduced by incomplete candidate set

As discussed in section 2.5.4, in some cases the true candidate is not include in the candidate set for junction squiggle X^k. If so, none of the candidate squiggles represents the expected shape of X^k, and the outputs of NanoSplicer will be unreliable. After constructing the empirical distributions, I tested how well the NanoSplicer certainty quantifies the potential uncertainty introduced by incomplete candidate set. Here, incomplete candidate set means the true candidate is not included in the candidate set. I first randomly split 11585 junction squiggles into group H1 and H0 with the same setting as the previous

section. For each junction squiggle in both groups, I calculated the NanoSplicer certainty according to the model in section 2.5.4 formula 2.12. In the formula, both $Pr(H^k = 1)$ and $Pr(H^k = 0)$ were set as 0.5, which matched the proportion of junctions in H1 and H0, respectively. I calculated the proportion of junction squiggles that came from H1 and H0 for 10 groups of junction squiggles (grouped according to the NanoSplicer certainty, Figure 3.5B). The result shows the junction squiggles with higher NanoSplicer certainty were more likely to come from H1, which means that the NanoSplicer certainty provides a good measure of the uncertainty introduced by incomplete candidate set.

3.2.2 NanoSplicer certainty indicates the potential uncertainty when the true candidate is known to be included

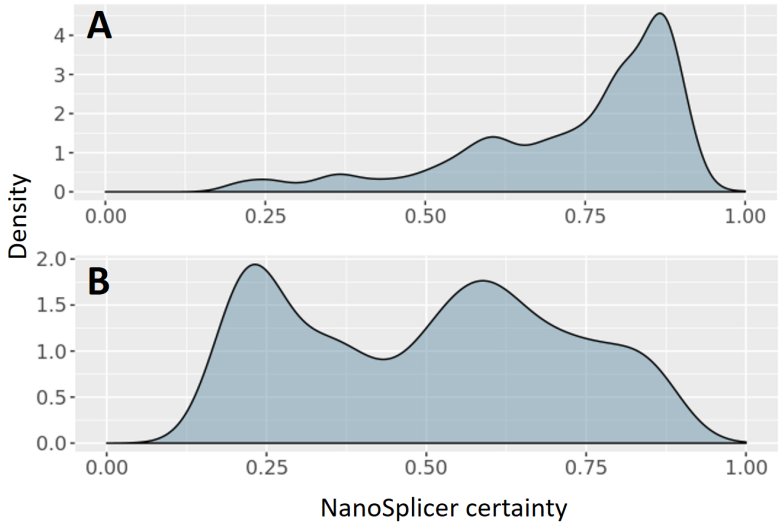

Figure 3.6:
Density of NanoSplicer certainty for junction squiggles whose splice sites are correctly (Figure A) or incorrectly (Figure B) identified by NanoSplicer.

In section 2.5.4, $H^k = 0$ is defined as the candidate squiggles do not contain the expected shape of the junction squiggle X^k. It occurs not only when the candidate set is incomplete. It could be also caused by the potential error in obtaining junction squiggle from Tombo. In this section, I investigated how NanoSplicer certainty links to NanoSplicer's performance when the true candidates were known to be included. I calculated NanoSplicer certainty for all 11585 squiggles, whose true candidates have been included. Figure 3.6 shows the difference in the distributions of NanoSplicer certainty for junction squiggles whose splice site were correctly or incorrectly identified by NanoSplicer. It suggests that the NanoSplicer certainties tend to be lower in the failed cases, which means NanoSplicer

is more likely to give wrong results for junction squiggles with low NanoSplicer certainty.

3.3 Filtering out the junction squiggles with low NanoSplicer certainty improves the reliability of NanoSplicer output

According to the design of the model, low NanoSplicer certainty means none of the candidate squiggles matches well with a junction squiggle. This can happen for multiple reasons, along with the situation outlined previously, it can also occur when the squiggle has low signal-to-noise ratio. No matter the cause of low NanoSplicer certainty, the final squiggle-to-candidate assignment is less reliable.

Hence, it is reasonable not to apply NanoSplicer on junction squiggles when the NanoSplicer certainty is too low. Figure 3.7 shows the proportion of remaining junction squiggles that were assigned to the true candidate by NanoSplicer after applying different thresholds on NanoSplicer certainty, as well as the remaining proportion of junction squiggles. The result shows that after filtering out the junction squiggles with low NanoSplicer certainty, a larger proportion of junction squiggles were assigned to the true candidate. There is also a line showing the number of squiggles that have been discarded after filtering. Since the sequencing depth for nanopore data is usually low, an overly stringent threshold would not be recommended.

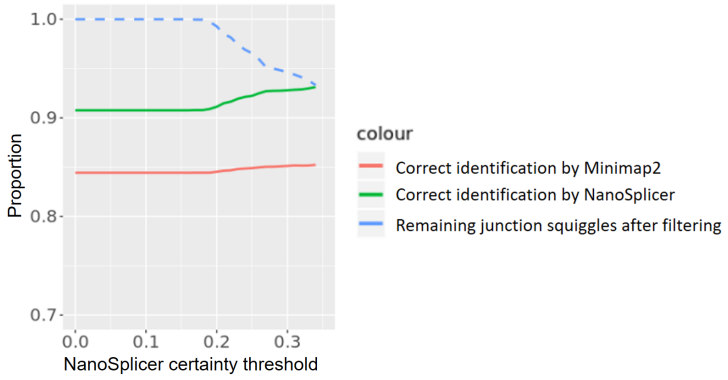

Figure 3.7: Filter out junction squiggles with low NanoSplicer certainty
Junction squiggles with NanoSplicer lower than a certain threshold were discarded. The solid lines show the proportion of junction squiggles/junctions within reads whose splice sites were correctly identified by NanoSplicer (green) and Minimap2 (red), after filtering with difference threshold. The dashed line shows the proportion of junction squiggles retained after filtering.

Discussion

All the current bioinformatics tools that perform splice site correction use the nanopore reads only. However, additional information about the correct splice site is contained within the squiggles. I developed a method called "NanoSplicer" to accurately identify the splice site using nanopore squiggles in addition to nanopore reads. The primary strategy for NanoSplicer is to find the most likely squiggle from a set of candidates generated from potential splice sites.

After testing NanoSplicer on a dataset with a known ground truth, relative to the raw mapping result (by minimap2), NanoSplicer successfully improved the splice site identification accuracy by 6.4%. It is worth noting that even though I made a lot of comparison between NanoSplicer and minimap2, the aim of developing NanoSplicer is not to build a mapper and compete with mappers like minimap2, but it is to offer a tool that can be used to improve exon junction identification. NanoSplicer actually uses minimap2 for the mapping step.

I have also defined a measure of NanoSplicer certainty, which has proved useful in indicating the level of confidence underpinning the decision made by NanoSplicer on a specific squiggle. In the data analysis section, I constructed the empirical distribution based on true splice sites, which is not known in real biological data analysis. These empirical distributions can still be constructed by assuming that the candidate with the highest assignment probability is the true candidate for most junction squiggles.

The development of NanoSplicer is yet to be completed. There are still a number of aspects of the method that can be improved. Although most labs choose cDNA for Nanopore transcript analysis, I will also test our method on direct RNA (dRNA) sequencing data. Unfortunately, the current pipeline doesn't perform very well on dRNA data. The speed RNA molecules traverse through the pore is much slower and more unstable than for DNA molecules, which makes the squiggle noisier and makes it hard for the DTW algorithm

to find the right location. Further tuning of the method is required to make the method robust enough for dRNA sequencing data. In addition, NanoSplicer currently uses uniform priors and . However, this may be not true in reality because true candidates will usually be included when the candidate set is generated in a proper way. More research on how to determine the prior is needed, for example, the prior can be learned from the data using an empirical Bayes method.

Last but not least, the current results were on the sequins RNA, which is synthetic RNA and might not reveal some biological characteristics of cellular RNA. It is better to validate the method on biological samples further. The next stage of the data analysis will be applied to samples that have been sequenced by both Nanopore and Illumina. There is no ground truth for biological samples, but the Illumina reads have much higher accuracy and could potentially provide information on the presence of minor splice sites and their relative quantity.

Research plans and research activities

5.1 Training

In my candidature year, I have attended following workshops/conferences:

- 2019 Winter School in Mathematical & Computational Biology (The University of Queensland, 1-5 July 2019): The winter school is to introduce hot topics in bioinformatics and computational biology to advanced undergraduate and postgraduate students and postdoctoral researchers.
- Victoria Cancer Bioinformatics Symposium 2019 (Peter MacCallum Cancer Centre, 16 Aug 2019): A student symposium focused on bioinformatics in cancer research.
- GIW/ABACBS 2019 (the University of Sydney, 9-11 Dec 2019): A conference broadly related to bioinformatics, computational biology, genomics, transcriptomics, proteomics, metabolomics, metagenomics, precision medicine and systems biology. I gave a presentation and fast-forward talk in the conference
- MIG scRNA-seq workshop (Melbourne Integrative Genomics(MIG), the University of Melbourne, 2-3 Oct 2019): In this workshop, I had opportunities to understand single-cell RNA sequencing (scRNAseq), and also gained practical scRNAseq data analysis experience using existing methods.
- MIG Seminar Series (MIG, University of Melbourne, fortnightly): A seminar series related to broad areas in bioinformatics, statistics etc.
- Funwip (MIG, University of Melbourne, weekly): It is a mixture of work-in-progress talks and a journal club with multiple research groups in MIG involved.

I will continue attending ABACBS, MIG seminar series and Funwip in the following years. In addition , I plan to attend following conferences/workshops:

- Genome informatics (Hinxton, UK, 17 sep 2020): A conference focuses on understanding the structure and the biology of genomes using large-scale approaches. It covers topics such as metagenomics, computational biology, transcriptomics etc.
- ONT London calling 2021 (Londan, UK, 2021): It is an annual conference dedicated to nanopore sequencing. The talks will focus on different applications of nanopore sequencing, as well as the bioinformatics methods.
- HMM study group(MIG, weekly) A study group launched by Dr. Heejung Shim that covers useful statistics methods such as Hidden Markov Model (HMM), Expectation-maximization (EM) algorithms, etc.

5.2 Research plan

Table 5.1: Research Plan

Research Plan		2020										2021	
		Mar-Apr		May-Aug				Sep-Dec				Jan-Feb	
Splice site detection	Apply NanoSplicer on reads from cellular RNA												
	Compare NanoSplicer to other splice site detection software												
	Write up and submit a manuscript for NanoSplicer												
	Make the NanoSplicer code and documentation available online												
	Enable NanoSplicer to analyse nanopore direct RNA sequencing data												
Upgrade NanoSplicer to NanoIsoform	Include the ability of isoform identification in NanoSplicer												
	Include the ability of isoform quantification in NanoSplicer												
Thesis Milestones	Second year report												

Table 5.2: Research Plan Continued

Research Plan		2021										2022			
		Mar-Apr		May-Aug				Sep-Dec				Jan-Apr			
Upgrade NanoSplicer to NanoIsoform	Include the ability of isoform quantification in NanoSplicer														
	Performance assessment of NanoIsoform (upgraded version of NanoSplicer)														
	Compare NanoIsoform with other software														
	Write up and submit a manuscript on NanoIsoform														
	Apply NanoIsoform in analysing differential expression between groups														
Thesis Milestones	Third year report														
	Thesis writing														

Bibliography

[1] Chun-Hao Su, Dhananjaya D, and Woan-Yuh Tarn. Alternative splicing in neurogenesis and brain development. *Front Mol Biosci*, 5:12, February 2018.

[2] Timothy W Nilsen and Brenton R Graveley. Expansion of the eukaryotic proteome by alternative splicing. *Nature*, 463(7280):457–463, January 2010.

[3] Auinash Kalsotra, Xinshu Xiao, Amanda J Ward, John C Castle, Jason M Johnson, Christopher B Burge, and Thomas A Cooper. A postnatal switch of CELF and MBNL proteins reprograms alternative splicing in the developing heart. *Proc. Natl. Acad. Sci. U. S. A.*, 105(51):20333–20338, December 2008.

[4] N A Faustino. Pre-mRNA splicing and human disease, 2003.

[5] Xavier Roca, Ravi Sachidanandam, and Adrian R Krainer. Intrinsic differences between authentic and cryptic 5' splice sites. *Nucleic Acids Res.*, 31(21):6321–6333, November 2003.

[6] Peter Stoilov, Eran Meshorer, Marieta Gencheva, David Glick, Hermona Soreq, and Stefan Stamm. Defects in pre-mRNA processing as causes of and predisposition to diseases. *DNA Cell Biol.*, 21(11):803–818, November 2002.

[7] Bushra Raj and Benjamin J Blencowe. Alternative splicing in the mammalian nervous system: Recent insights into mechanisms and functional roles. *Neuron*, 87(1):14–27, July 2015.

[8] Celine K Vuong, Douglas L Black, and Sika Zheng. The neurogenetics of alternative splicing, 2016.

[9] Eric T Wang, Rickard Sandberg, Shujun Luo, Irina Khrebtukova, Lu Zhang, Christine Mayr, Stephen F Kingsmore, Gary P Schroth, and Christopher B Burge. Alternative isoform regulation in human tissue transcriptomes, 2008.

[10] Nicolas M Bertagnolli, Justin A Drake, Jason M Tennessen, and Orly Alter. SVD identifies transcript length distribution functions from DNA microarray data and reveals evolutionary forces globally affecting GBM metabolism. *PLoS One*, 8(11):e78913, November 2013.

[11] Tamara Steijger, Josep F Abril, Pär G Engström, Felix Kokocinski, RGASP Consortium, Tim J Hubbard, Roderic Guigó, Jennifer Harrow, and Paul Bertone. Assessment of transcript reconstruction methods for RNA-seq. *Nat. Methods*, 10(12):1177–1184, December 2013.

[12] Laura H LeGault and Colin N Dewey. Inference of alternative splicing from RNA-Seq data with probabilistic splice graphs. *Bioinformatics*, 29(18):2300–2310, September 2013.

[13] David Deamer, Mark Akeson, and Daniel Branton. Three decades of nanopore sequencing. *Nat. Biotechnol.*, 34(5):518–524, May 2016.

[14] J J Kasianowicz, E Brandin, D Branton, and D W Deamer. Characterization of individual polynucleotide molecules using a membrane channel. *Proc. Natl. Acad. Sci. U. S. A.*, 93(24):13770–13773, November 1996.

[15] Alexander S Mikheyev and Mandy M Y Tin. A first look at the oxford nanopore MinION sequencer, 2014.

[16] Mohan Bolisetty, Gopinath Rajadinakaran, and Brenton Graveley. Determining exon connectivity in complex mRNAs by nanopore sequencing.

[17] Lin Liu, Yinhu Li, Siliang Li, Ni Hu, Yimin He, Ray Pong, Danni Lin, Lihua Lu, and Maggie Law. Comparison of next-generation sequencing systems. *J. Biomed. Biotechnol.*, 2012:251364, July 2012.

[18] Mehdi Kchouk, Jean Francois Gibrat, and Mourad Elloumi. Generations of sequencing technologies: From first to next generation, 2017.

[19] Christoph Bleidorn. Third generation sequencing: technology and its potential impact on evolutionary biodiversity research, 2016.

[20] nextgenseek. Comparing price and tech. specs. of illumina MiSeq, ion torrent PGM, 454 GS junior, and PacBio RS. http://nextgenseek.com/2012/08/comparing-price-and-tech-specs-of-illumina-miseq-ion-torrent-pgm-454-gs-junior-and-pacbio-rs/, August 2012. Accessed: 2020-2-7.

[21] Product comparison. http://nanoporetech.com/products/comparison. Accessed: 2020-2-7.

[22] Ryan R Wick, Louise M Judd, and Kathryn E Holt. Performance of neural network basecalling tools for oxford nanopore sequencing. *Genome Biol.*, 20(1):129, June 2019.

[23] Nobuaki Kono and Kazuharu Arakawa. Nanopore sequencing: Review of potential applications in functional genomics. *Dev. Growth Differ.*, 61(5):316–326, June 2019.

[24] Alberto Magi, Roberto Semeraro, Alessandra Mingrino, Betti Giusti, and Romina D'Aurizio. Nanopore sequencing data analysis: state of the art, applications and challenges. *Brief. Bioinform.*, 19(6):1256–1272, November 2018.

[25] DNA sequencing — oxford nanopore technologies. http://nanoporetech.com/applications/dna-nanopore-sequencing. Accessed: 2020-2-8.

[26] Franka J Rang, Wigard P Kloosterman, and Jeroen de Ridder. From squiggle to basepair: computational approaches for improving nanopore sequencing read accuracy. *Genome Biol.*, 19(1):90, July 2018.

[27] T Z Butler, M Pavlenok, I M Derrington, M Niederweis, and J H Gundlach. Single-molecule DNA detection with an engineered MspA protein nanopore, 2008.

[28] Elizabeth A Manrao, Ian M Derrington, Andrew H Laszlo, Kyle W Langford, Matthew K Hopper, Nathaniel Gillgren, Mikhail Pavlenok, Michael Niederweis, and Jens H Gundlach. Reading DNA at single-nucleotide resolution with a mutant MspA nanopore and phi29 DNA polymerase, 2012.

[29] Gerald M Cherf, Kate R Lieberman, Hytham Rashid, Christopher E Lam, Kevin Karplus, and Mark Akeson. Automated forward and reverse ratcheting of DNA in a nanopore at 5-Å precision, 2012.

[30] Peter Sarkozy, Ákos Jobbágy, and Peter Antal. Calling homopolymer stretches from raw nanopore reads by analyzing k-mer dwell times, 2018.

[31] Eli Koren, Galit Lev-Maor, and Gil Ast. The emergence of alternative 3 and 5 splice site exons from constitutive exons, 2007.

[32] nanoporetech. nanoporetech/pinfish. https://github.com/nanoporetech/pinfish. Accessed: 2020-2-9.

[33] Alison D Tang, Cameron M Soulette, Marijke J van Baren, Kevyn Hart, Eva Hrabeta-Robinson, Catherine J Wu, and Angela N Brooks. Full-length transcript characterization of SF3B1 mutation in chronic lymphocytic leukemia reveals downregulation of retained introns.

[34] Matthew Loose, Sunir Malla, and Michael Stout. Real-time selective sequencing using nanopore technology. *Nat. Methods*, 13(9):751–754, September 2016.

[35] James M Ferguson and Martin A Smith. SquiggleKit: a toolkit for manipulating nanopore signal data. *Bioinformatics*, 35(24):5372–5373, December 2019.

[36] Heng Li. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, 34(18):3094–3100, 2018.

[37] M Burset. Analysis of canonical and non-canonical splice sites in mammalian genomes, 2000.

[38] James M Ferguson and Martin A Smith. Squigglekit: A toolkit for manipulating nanopore signal data. *Bioinformatics*, 35(24):5372–5373, 2019.

[39] Re-squiggle algorithm — tombo 1.5 documentation. https://nanoporetech.github.io/tombo/resquiggle.html. Accessed: 2020-2-9.

[40] 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.

[41] Eamonn Keogh and Chotirat Ann Ratanamahatana. Exact indexing of dynamic time warping. *Knowledge and information systems*, 7(3):358–386, 2005.

[42] Simon A Hardwick, Wendy Y Chen, Ted Wong, Ira W Deveson, James Blackburn, Stacey B Andersen, Lars K Nielsen, John S Mattick, and Tim R Mercer. Spliced synthetic genes as internal controls in RNA sequencing experiments. *Nat. Methods*, 13(9):792–798, September 2016.