Searching and mapping genomic subsequences in nanopore raw signals through novel dynamic time warping algorithms

Renmin Han[1], Sheng Wang[1,*], Xin Gao[1,*]

[1]King Abdullah University of Science and Technology (KAUST), Computational Bioscience Research Center (CBRC), Computer, Electrical and Mathematical Sciences and Engineering (CEMSE) Division, Thuwal, 23955-6900, Saudi Arabia.

Abstract

Nanopore sequencing is a promising technology to generate ultra-long reads based on the direct measurement of electrical current signals when a DNA molecule passes through a nanopore. These ultra-long reads are critical for detecting large structural variations in the genome. However, it is challenging to use nanopore sequencing to identify single nucleotide polymorphisms (SNPs) or other modifications such as methylations, especially at a low sequencing coverage, due to the high error rate in the base-called reads. It is possible to correct the base-calling error through the subsequence search by mapping a SNP-containing genomic region to the long nanopore raw signal sequences that contain this region and taking consensus of these signals. Nevertheless, the ultra-long raw signals and an order of magnitude difference in the sampling speed between the two sequences make the traditional algorithms infeasible to solve the problem.

Here we propose two novel algorithms, the direct subsequence dynamic time warping for nanopore raw signal search (DSDTWnano) and the continuous wavelet subsequence dynamic time warping for nanopore raw signal search (cwSDTWnano), to enable the direct subsequence searching and exact mapping in nanopore raw signals. The proposed algorithms are based on the idea of subsequence-extended dynamic time warping and directly operate on the raw signals, without any loss of information. DSDTWnano could ensure an output of highly accurate query results and cwSDTWnano is the accelerated version of DSDTWnano, with the help of seeding and multi-scale coarsening of signals that are based on continuous wavelet transform. Furthermore, a novel error function is proposed to specify the mapping accuracy between a genomic sequence and an electrical current signal sequence, which may serve as the standard criterion for further genome-to-signal mapping studies. Comprehensive experiments on three real-world nanopore datasets (human and lambda phage) demonstrate the efficiency and effectiveness of the proposed algorithms. Finally, we show the power of our algorithms in SNP detection under a low coverage (20×) on E. coli, with >95% detection rate.

Our program is available at `https://github.com/icthrm/cwSDTWnano.git`.

*All correspondence should be addressed to Xin Gao (xin.gao@kaust.edu.sa) and Sheng Wang (sheng.wang@kaust.edu.sa).

1 Introduction

Nanopore sequencing is a rapidly developing technology that is able to generate 10-50k ultra-long reads in real time on a portable instrument at low-cost [1]. The key innovation of nanopore sequencing resides in the pore chemistry, which directly measures the electrical current signal (denoted as the *raw signal*) when a single-strand DNA passes through the nanopore [2]. Long reads are extremely valuable to provide long-distance information on genome or transcriptome, which simplifies the genome/transcriptome assembly and enables the detection of large structural variations [3]. However, it is challenging to use nanopore sequencing to detect single nucleotide polymorphisms (SNPs), which are the key for gene-disease diagnoses, association studies, ancestry inference, drug design, and linked region detection [4]. The main reason is due to the higher error rate (more than 10%) in the reads generated by nanopore sequencing compared to 1% error rate for 2nd generation sequencing technologies that generate short seeds [5].

According to recent studies [5], there are two sources of errors in nanopore sequencing: (i) the inherent error that comes from the low signal-to-noise ratio in the raw signals during sequencing; and (ii) the base-calling error that comes from the process of translating the raw signals into a DNA sequence by machine learning models, which depends heavily on the datasets that are used to train the parameters of the models. It has been shown that the training data (such as the type of species) could bias the model's actual performance when being applied on new species [5].

Nanopore base-calling is often hampered by mutations or modifications such as methylations, of which the non-standard information is kept in the raw signals. Thus, [6] utilized raw signals to detect DNA methylations via a hidden Markov model (HMM) model. [7] proposed a method for *de novo* detection of DNA modifications without any model training. The key component of these methods is the end-to-end mapping between the reference genomic sequence and the raw signal sequence, which is firstly translated to a read through base-calling. However, such an approach still inevitably introduces base-calling errors and the information loss is irreversible. On the contrary, an alternative approach has been proposed by firstly translating the reference sequence to expected signals via the official k-mer pore model and then aligning the two signal sequences [8].

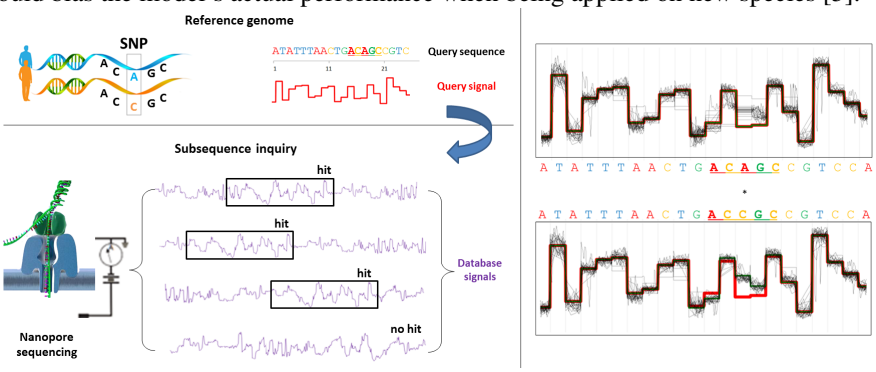

Figure 1: Left part: the SNP detection problem could be formulated as a subsequence inquiry problem at the signal level. Here the query sequence is the *expected signals* transferred from the SNP-containing genomic region by the official pore model, whereas the database contains the raw electrical current signal sequences (i.e. *raw signals*) directly measured from nanopore sequencing. Right part: the aligned raw signals upon two query sequences with a SNP (here is $A \rightarrow C$). A clear difference between the raw signals (shown in black squiggle curves) and the expected signals (shown in red and green curves for reference and mutated sequences, respectively) indicates a strong evidence of SNP.

In practice, biologists often have an idea of where approximately (e.g., in which genes) SNPs take place and what need to be determined are their exact locations and the nucleotide polymorphisms. With the help of nanopore sequencing, the problem of SNP detection could be formulated as a subsequence search problem. The inputs are (i) a candidate reference genomic region in which SNPs exist, and (ii) a database of nanopore raw signals for the corresponding species. Our aim is to detect the raw signal sequences that contain the reference genomic region and map them to the reference region to discover the SNPs (see Fig. 1). Most of the current methods make use of the base-called reads, which are error-prone and loss information from the raw signals [5]. On the other hand, although there are various methods for subsequence searching and mapping of genomic sequences [9, 10, 11], there is no applicable method, to our knowledge, for subsequence search of nanopore signal sequences. There are three potential reasons for the absence of such methods. First, the length of raw signal sequences is very long, ranging from 100k to 500k time points. Second, there is an order of magnitude difference between the length of raw signals and that of expected signals, where the former is often 8-10 times larger than the latter. Third, the signal sequences are composed of real values, whereas the genomic sequences are composed of discrete letters.

In this paper, we propose two novel algorithms to enable the direct subsequence search and exact mapping in the nanopore raw signal database (i.e., local genome-to-signal search). The proposed algorithms are based on the idea of subsequence Dynamic Time Warping (sDTW) and directly operate on the nanopore raw signals without any loss of information in data translation. The first algorithm is Direct Subsequence Dynamic Time Warping for nanopore raw signal search (DSDTWnano), which is based on a subsequence-extended DTW followed by boundary constrained

mapping search. DSDTWnano could ensure an output of highly accurate query results and run in an $O(MN)$ time complexity, where M is the query length and N is the raw signal length. The second algorithm is continuous wavelet Subsequence DTW for nanopore raw signal search (cwSDTWnano), which is an accelerated version of DSDTWnano with the help of seeding and multi-scale coarsening based on continuous wavelet transform (CWT). For a typical similarity search with 4000bp-long query and a nanopore raw signal sequence of 2×10^5 time points, cwSDTWnano could finish the search in 600 ms. Furthermore, a novel error measurement is proposed to specify the mapping accuracy between a query genomic sequence and a raw signal sequence. Comprehensive experiments on three real-world nanopore datasets demonstrate the efficiency and effectiveness of the proposed algorithms. Finally, we demonstrate the power of our algorithms in SNP detection under a low coverage (20×), with >95% detection rate.

2 Preliminaries

In this section, we first define the subsequence inquiry problem in nanopore sequencing. We then introduce DTW and sDTW which are the foundations to our proposed algorithms. Finally, we show by a case study that why the traditional sDTW cannot be applied to the local genome-to-signal search (i.e., a specific case of subsequence inquiry) and how to solve this problem by resampling techniques.

2.1 Subsequence inquiry problem in nanopore sequencing

Given a query genomic sequence and a raw signal database, as shown in Fig. 1, the subsequence inquiry problem is to detect the segments of raw signals in the database that are similar to the query genomic sequence (the *hit signals*). On the contrary, the raw signals with no high-similarity segment to the query sequence are denoted as *non-hit signals*. Such a problem could also be denoted as local genome-to-signal search (short repeat sequence and homopolymers are not considered here). Formally, let $X = (x_1, x_2, \cdots, x_N)$ be a raw signal sequence, and $Y = (y_1, y_2, \cdots, y_M)$ be the expected query signal sequence (abbr. *query signal*) that is translated from the query genomic sequence by the pore model, where $M < N$. Note that the sampling rate of X is about 8 to 9 times higher than that of Y. Our aim is to find a subsequence $X[t_s : t_e] = (x_{t_s}, \cdots, x_{t_e})$ of X ($1 \le t_s < t_e \le N$) that minimizes the distance measurement between Y and all possible subsequences of X, i.e.,

$$E(t_s^{opt}, t_e^{opt}) = \underset{(t_s,t_e):1\le t_s<t_e\le N}{\arg\min} \ \mathrm{Dist}(Y, X[t_s : t_e]). \tag{1}$$

2.2 Dynamic time warping (DTW) and subsequence DTW (sDTW)

In time series analysis, dynamic time warping (DTW) is one of the algorithms for measuring similarity between two temporal sequences that vary in speed. Given a query sequence $Y = (y_1, y_2, \cdots, y_M)$ and a database sequence $X = (x_1, x_2, \cdots, x_N)$, the DTW distance $\mathrm{Dist}(Y, X)$ is defined iteratively:

$$\begin{cases} \mathrm{Dist}(Y, X) = D(M, N); \\ D(i, j) = \|y_i - x_j\| + \min\{D(i-1, j), D(i, j-1), D(i-1, j-1)\}; \\ D(0, 0) = 0; D(i, 0) = D(0, j) = \infty. \end{cases} \tag{2}$$

It can be seen that the DTW distance is able to be exactly solved by dynamic programming in $O(MN)$ time, resulting in the globally optimal alignment. For local genome-to-signal search, a naive solution is to open a sliding window for each time point in the raw signal and calculate the DTW distance for each sliding window, which would result in $O(M^2N)$ time complexity. We observe that the DTW distance is able to distinguish the regions of high similarity in the raw signal sequence (Fig. 2(B)), whereas the Euclidean distance is not able to do so (Fig. 2(A)). However, the high computational complexity hampers the application of this approach.

The problem of finding optimal subsequences can be solved by a variant of DTW, namely subsequence DTW (sDTW) [12]. In particular, we pad the query to $Y' = (y_0, y_1, y_2, \cdots, y_M)$, and define $\|y_0, x_i\| = 0$ for all x_i. Subsequently, the minimum distance $\mathrm{Dist}(X[t_s : t_e], Y)$ could be derived as follows:

$$\begin{cases} \mathrm{Dist}(X[t_s : t_e], Y') = D(t_e, M) = \min(D(t, M)) \\ D(t, j) = \|x_t - y_j\| + \min\{D(t-1, j), D(t, j-1), D(t-1, j-1)\}; \\ D(t, 0) = 0, D(0, i) = \infty. \end{cases} \tag{3}$$

After calculating the entire distance matrix, the optimal mapping path could be traced from the end point t_e in $D(t, M)$ to the start point t_s. Obviously, the time complexity of this algorithm is $O(MN)$.

However, it should be noted that sDTW is sensitive to gaps in the alignment, which makes it infeasible to the local genome-to-signal search problem in naopore sequencing due to an order of magnitude difference in the sampling speed (Fig. 2(C)). Fortunately, this issue can be solved by resampling X to generate a compressed signal sequence, X', with a similar sampling rate as Y (Fig. 2(D)).

2.3 An example of local genome-to-signal search in nanopore raw signals

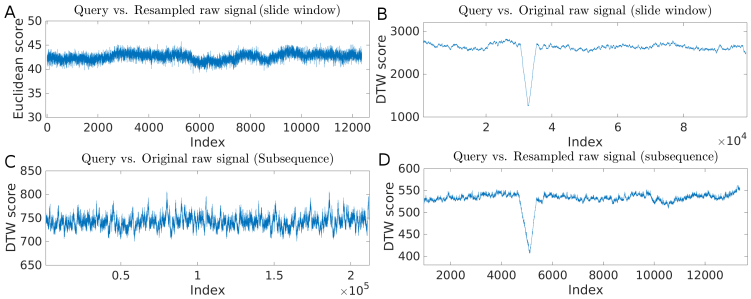

Figure 2: An example showing the distinguishing power of different distance measures and different variants of DTW on subsequence search. (A) Subsequence search by the Euclidean distance between the query Y and the resampled signal sequence X'; (B) Subsequence search by the DTW distance between the query Y and the raw signal sequence X; (C) Subsequence search by sDTW between the query Y and the raw signal sequence X; (D) Subsequence search by sDTW between the query Y and the resampled signal sequence X'.

Here we design a simple experiment to test different distance measurements and different variants of DTW on subsequence inquiry of nanopore signals. As the raw signals have on average 8 to 9 times of redundant sampling rate, we use the FIR (finite impulse response filter) resampling technique [13] to generate a 8-times compressed signal sequence X' from X. A query sequence with 1000 nucleotides (Y) and a nanopore raw signal sequence with ~100,000 time points (X), which contains the query sequence, are selected to demonstrate the results (Fig. 2).

As shown in Fig. 2(A), the Euclidean distance has no distinguishing power to identify the raw signal subsequence that is similar to the query sequence. On the contrary, Fig. 2(B) shows that the DTW distance could pick up the region correctly, but the sliding window based search strategy took half an hour to get the result. The sDTW method fails to identify the region (Fig. 2(C)), which might be due to the massive amount of redundant sampling in the raw signals. Finally, Fig. 2(D) shows that sDTW is able to detect a shape peak on the compressed signal sequence.

3 Materials and Methods

Here, two novel algorithms are proposed for direct subsequence searching and mapping in nanopore raw signals. The first algorithm is Direct Subsequence Dynamic Time Warping for nanopore raw signal search (DSDTWnano). DSDTWnano is based on an effective subsequence DTW and the following boundary constrained DTW to ensure the accurate mapping between the query sequence and the raw signal sequence. The second algorithm is continuous wavelet Subsequence DTW for nanopore raw signal search (cwSDTWnano). cwSDTWnano is an accelerated version of DSDTWnano, which dramatically reduces the execution time by utilizing the concept of seed and adopting the strategy of multi-scale analysis.

3.1 Direct subsequence dynamic time warping for nanopore raw signal search

The main difficulty to apply subsequence DTW on the nanopore raw signal data is the scale difference between the query and the raw signal sequences. We propose to resolve this issue by resampling the raw signal sequence first, aligning the resampled signals to the query, remapping the warping path of the resampled signals to the original ones, and finally refining it by constrained DTW. Because the highly similar region will result in a sharp peak, an early stop condition could be introduced to save runtime when we calculate DTW distance along the nanopore raw signal.

We thus propose a simple algorithm, DSDTWnano (Algorithm 1), where DSDTW($\cdot$) is the subsequence dynamic time warping with an early stop condition and will report the end location of high similarity, Resampling($\cdot$) is the FIR-based resampling to compress the nanopore raw signals [13], PathTrackback($\cdot$) is a function that recursively searches the match paths between X' and Y that starts from t_e, ReMapIndex($\cdot$) is a context-dependent constraint generation from a coarse path W_{coarse} with a window size of r, cDTW($\cdot$) is the constrained dynamic time warping [14] and s_{base} is the estimation of raw signal's sampling rate (usually around 8). Because the subsequence DTW has the complexity of $O(MN')$ and the constrained DTW has the complexity of $O(rM)$ ($N' \approx N/s_{base}$, which is the length of resampled signals), the overall complexity of DSDTWnano is $O(\frac{1}{s_{base}}MN)$.

3.2 Continuous wavelet subsequence DTW for nanopore raw signal search

Compared with the sliding-window based DTW, DSDTWnano has a dramatic improvement in computational complexity. However, it is still slow when the query signal or the raw signal sequence is very long. To accelerate the efficiency while keeping the effectiveness, we develop cwSDTWnano, which includes several techniques to further speed up the subsequence similarity search on nanopore raw signals: seeding, pre-filtering, and multi-scale search based on continuous wavelet transform (CWT).

In particular, cwSDTWnano starts from seed search on the compressed raw signals. Based on the mapping paths of the seeds, the signal sequences with no high-similarity segment (i.e., non-hit signals) are filtered out. For the candidate signal sequences that pass the filter, a low-resolution wavelet transform is imposed on the long nanopore signal and the query signal sequences to highly compress the information, which are utilized to generate the coarse path with the help of seeds. Finally, with the multi-scale analysis of CWT, the mapping path between the query signal sequence and the raw signal sequence is calculated recursively from a lower-resolution projection to a higher-resolution one.

Algorithm 1: DSDTWnano

Input: long reference X, query Y, scale s_{base} and boundary r

SubProcedure DSDTW(X, Y)
  Pad Y to $Y' = (y_0, y_1, \cdots, y_M)$;
  $D_{min} \leftarrow \infty$, $t_e \leftarrow 0$, $t \leftarrow 0$;
  while $t \leq N$ **do**
    $D(t, 0) \leftarrow 0$;
    for $j = 1$ ***to*** M **do**
      $d \leftarrow \|x_t - y_j\|$;
      $D(t, j) =$
        $d + \min\{D(t-1, j), D(t, j-1), D(t-1, j-1)\}$;
    end
    if $D_{min} > D(t, M)$ **then**
      $D_{min} = D(t, M)$, $t_e = t$;
    end
    Update $\epsilon = \text{avg}(D(\cdot, M)) - 5\text{dev}(D(\cdot, M))$;
    if $t - t_e \geq M$ *and* $D_{min} < \epsilon$ **then**
      break;
    end
    $t \leftarrow t + 1$;
  end
  return t_e.
Procedure DSDTWnano(X, Y, s_{base}, r)
  $X' = \text{Resampling}(1/s_{base}, X)$, i.e., $X' = (x'_1, \cdots, x'_{N'})$;
  $t_e = \text{DSDTW}(X', Y)$;
  $W_{coarse} = \text{PathTrackback}(t_e, X', Y)$;
  $B = \text{ReMapIndex}(W_{coarse}, r)$;
  $W_{fine} = \text{cDTW}(X, Y, B)$;
  return W_{fine}.

3.2.1 Seeds with *minimal length*

In genomic read mapping, the k-long subsequences (i.e., k-mers) in a query sequence are often used as a quick indicator of whether and where the reference contains the query. These k-mers are called "seeds" and their inquiry is usually done through hashing. Because of the high noise and non-stable sampling rates in nanopore sequencing, it is difficult to build such a k-mer hash function in the similarity search of the query signal. However, we still can use the idea of "seed" to quickly determine the range where the query signal locates in the raw signals.

Our observation is that if a query signal exceeds a certain length, we could detect this signal in the long raw signal sequence without ambiguity, and this certain length is denoted as the *minimal length*. An experiment is presented to show how the length of the query affects the similarity search. As discussed in Section 2.3, the subsequence in the compressed raw signals with the highest similarity to the query signal will result in the minimum DTW distance, which behaves as a sharp peak. Fig. 3(A) shows that a 32bp- or 64bp-long query signal cannot determine a unique result because there is no distinguishable peak of the DTW distance among different locations. On the contrary, a query with 96bp or 128bp length is able to detect a clear sharp peak. Two reasons may explain why a very short query fails: (i) the noise in the raw signal degenerates the DTW distance of the true result, and (ii) there exist multiple similar subsequences in the raw signal sequence.

Through comprehensive experiments, we find that a length of 128bp is enough for a short query to be detected in the raw signals. We also found that if the distributions of the electrical current values in raw signals are given, we may infer the theoretical *minimal length* from the given distribution, which shows the existence of the *minimal length* in any nanopore system (a brief proof is given in Section S1). Therefore, we denote a short segment in a query signal with length at least *minimal length* as a *seed*.

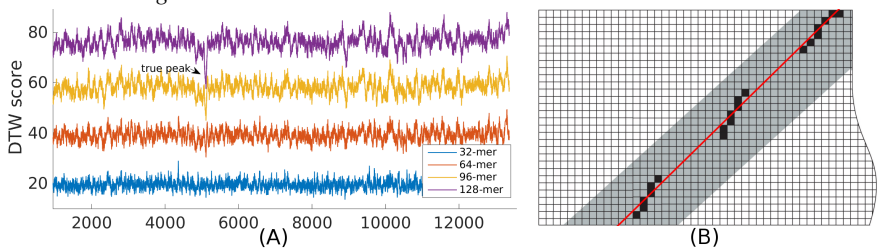

Figure 3: (A) The change of the DTW distance with different query lengths. (B) Linear relationship of the mapped path of 3 short seeds that are extracted from a long query sequence.

3.2.2 Filtering non-hit signals by seeds

Given a long query signal sequence ($\geq$ 1000), it is possible to utilize the *seeds* to filter raw signals with no high-similarity segment, which will significantly reduce the total query time. The key observation is that if a query sequence has a highly similar region in the compressed raw signal sequence, linear ordered seeds on the query sequence will also have a linear relationship to the hit regions in the compressed signal sequence (Fig. 3(B)). On the contrary, if the query sequence does not have a highly similar region in the compressed signal, no seeds will be detected or the seeds

will not follow a linear order. Based on this observation, a filtering operation is developed to quickly exclude those non-hit signals:

1) Select a set of segments $\{Q_i\}_{i=1,\cdots,K}$ from the query signal Y as the *seeds*;
2) For each *seed*, search in the compressed signal sequence X' by sDTW(·) to get the local mapping with the highest similarity;
3) Trackback from the endpoint of the mapping to get the mapping path for each *seed*;
4) Make a linear regression based on the mapping paths of these *seeds* and check their consistency;
5) If the consistency is weak, stop the process.

With the filtering operation, if the linear relationship of the seeds is violated, we can stop the search process to save time. If we select K seeds with length L, the total cost for a raw signal sequence with N time points is $O(\frac{N}{s_{base}}KL)$, in which both K and L are small numbers.

3.2.3 Multi-scale search based on continuous wavelet transform

When handling long signal sequences, multi-scale analysis has been widely used to reduce the runtime [15, 16, 17], and CWT has been adopted to preserve the feature information [18]. Recently, a multi-scale analysis with CWT has been proposed for global end-to-end mapping of two ultra-long signals [8]. Here we apply a similar idea to the genome-to-signal subsequence search problem.

Multi-scale representation by CWT: Following [8], a nanopore signal sequence could be transformed by CWT with different scales a to capture multi-scale information (a brief introduction of CWT is given in Section S2). Then, a feature extraction procedure can be carried out (denoted as PickPeaks(·)) to reduce the length of a raw signal X under different scales a:

1) Obtain the spectra CWT(X, a);
2) Normalize CWT(X, a) based on Z-score normalization;
3) Extract peaks and nadirs from each spectrum as the feature sequence.

The length of a signal sequence could be dramatically reduced by more than a times for a nanopore raw signal sequence.

Coarse path Generation: As introduced in Section 3.2.2, a number of *seeds* are used and their mapping paths with the compressed signal sequences are recorded. These short mapping paths can be used as anchors in the construction of the coarse mapping path between the query sequence and the raw signal sequence using the lowest resolution transform (i.e., with maximal level coarsening scale) from CWT:

1) Given the query sequence Y with length M, get the maximal level coarsening scale $a = \log_2(M) - 2$;
2) Get the feature signals for both CWT(X', a) and CWT(Y, a);
3) Run the subsequence DTW on the feature signals and get all the paths;
4) Find out the coarse path that covers the seeds;
5) Combine both the seeds and the coarse path to generate a more detailed path.

Then, the generated coarse mapping path is fed into cwDTW [8] to determine the final mapping.

The continuous wavelet subsequence DTW: Algorithm 2 shows cwSDTWnano, where cwDTW(·) is the continuous wavelet-based multi-level DTW [8], SelectSeeds(·) is the procedure to get K segments with length L from Y, CheckFalse(·) is the filtering of false alignment described in Section 3.2.2, ReMapIndex(·) is a context-dependent constraint generation from a coarse path W_{coarse} with a window size of r, CWT(·) is the continuous wavelet transform and PickPeaks(·) is the procedure to get feature sequence [8], CoarsePath(·) is the coarse path generation procedure described in the previous paragraph and cDTW(·) is the constrained DTW [14]. We notice that the false filtering procedure has a complexity of $O(\frac{N}{s_{base}}KL)$ and the procedure of cwDTW(·) is bounded within $O(N \log N)$, the overall complexity for Algorithm 2 is $O(\frac{N}{s_{base}}KL + N \log N)$, which has an obvious advantage when the signal length increases.

4 Experiments and Results

To demonstrate the efficiency and effectiveness of the two proposed local genome-to-signal search (abbr. local search) algorithms, we comprehensively evaluate the performance of DSDTWnano and cwSDTWnano on two real nanopore datasets from human and lambda phage. Furthermore, a successful SNP detection case study was presented using the dataset from E. coli with a low sequencing coverage.

4.1 Datasets

Three nanopore sequencing datasets are used in our experiments, among which the first (human) and second (lambda phage) are used to evaluate the accuracy of our proposed local search algorithms, and the third (E. coli) is used to demonstrate the power of our algorithms on hit signal discrimination and low-coverage SNP detection.

The first dataset is a subset of the public available human data, which comes from human chromosome 21 from the Nanopore WGS Consortium [19] and contains 6318 sequenced reads. The samples in this dataset were sequenced from the NA12878 human genome reference on the Oxford Nanopore MinION using 1D ligation kits (450 bp/s) with R9.4 flow cells[1]. We denote this dataset as the Human21 database.

The second and third datasets are from the genome of lambda phage and E. coli, respectively. These two datasets were all prepared and sequenced at the University of Queensland by Prof. Lachlan Coin's lab. The lambda phage dataset contains 27004 reads and the E. coli dataset contains 27608 reads. The samples were sequenced on the MinION device with 1D protocol on R9.4 flow cells (FLO-MIN106 protocol). We denote these two datasets as the Lambda phage database and the E. coli database, respectively. Specifically, E. coli has a relatively low coverage (20×).

To evaluate the accuracy of the algorithms, we created a subset by randomly sampling 3000 reads from Human21 and Lambda phage[2]. The average length of the DNA sequences in the sampled datasets is 7890 and 8461 for Human21 and Lambda phage, and the average length of the nanopore raw signal sequences is 65,947 and 69,715, respectively.

Algorithm 2: cwSDTWnano

Input: long reference X, query Y, scale s_{base}, seed number K, seed length L and boundary r

SubProcedure cwDTW(X, Y, W_S, S, r)
  Initialization: $s \leftarrow S - 1$;
  while $s \geq 1$ **do**
    $X^s = \text{CWT}(X, 2^{s-1})$, $P_{X^s} = \text{PickPeaks}(X^s_p)$;
    $Y^s = \text{CWT}(Y, 2^{s-1})$, $P_{Y^s} = \text{PickPeaks}(X^s_g)$;
    $B_s = \text{ReMapIndex}(W_{s+1}, r)$;
    $W_s = \text{cDTW}(P_{Y^s}, P_{X^s}, B_s)$;
    $s \leftarrow s - 1$;
  end
  return $W_{s=1}$.
Procedure cwSDTWnano(X, Y, s_{base}, s, L, r)
  $X' = \text{Resampling}(1/s_{base}, X)$, i.e., $X' = (x'_1, \cdots, x'_{N'})$;
  $\{Q_i\}_{i=1,\cdots,K} = \text{SelectSeeds}(Y, K, L)$;
  $\{W_i\}_{i=1,\cdots,K} \leftarrow 0$;
  foreach Q_i **do**
    $t_e = \text{DSDTW}(Q_i, Y)$,
     $W_i = \text{PathTrackback}(Q_i, X', Y)$;
  end
  if *CheckFalse*($\{W_i\}$) = ***true*** **then**
    return $\varnothing$;
  end
  $S = \log_2(M) - 2$;
  $X'^S = \text{CWT}(X', 2^S)$, $Y^S = \text{CWT}(Y, 2^S)$;
  $W_{coarse} = \text{CoarsePath}(X'^S, Y^S, \{W_i\})$;
  $W_{s=1} = \text{cwDTW}(X', Y, W_{coarse}, S, r)$;
  $B = \text{ReMapIndex}(W_{s=1}, r)$;
  $W_{fine} = \text{cDTW}(X, Y, B)$;
  return W_{fine}.

4.2 Evaluation criteria

4.2.1 Edit mapping error of a local search algorithm

Suppose the reference genome is known, we may use the edit mapping error to evaluate the difference between the mapping path generated by a local genome-to-signal search algorithm and the global mapping path.

Specifically, given a nanopore raw signal sequence, as we know the reference genome, it is possible to find the corresponding genomic region to the raw signals [11]. Then, the global mapping path W' between the genomic region and the raw signal sequence can be derived by the original DTW or other dynamic programming algorithms [8, 7].

Thus, for a genomic region $G = g_1 g_2 \cdots g_L$ and its corresponding raw signal sequence $R = r_1 r_2 \cdots r_N$, the accuracy of the mapping path W generated by a local search algorithm is defined as:

$$\text{emError}(W, W') = \frac{1}{L} \sum_{i=1}^{L} \frac{\text{EditDist}(\text{signal}_W(g_i), \text{signal}_{W'}(g_i))}{\text{L}(\text{signal}_{W'}(g_i))}, \tag{4}$$

where $\text{signal}_x(g_i)$ returns the set of signal indexes $\{r_j\}$ that corresponds to the query sequence position g_i from a certain mapping path $x \in \{W, W'\}$. This is because on average, each nucleotide corresponds to 8 to 9 signals in the raw signal sequence due to the redundant sampling in nanopore. EditDist(·) is the edit distance and L(·) is size of the signal index set.

For example, if we have a query $G = g_1 g_2 g_3$ with $L = 3$. Suppose its local mapping path W is $\{(10, 1), (11, 1), (12, 1), (13, 2), (14, 2), (15, 2), (16, 3), (17, 3)\}$, and the global mapping path W' is $\{(11, 1), (12, 1), (13, 2), (14, 2), (15, 2), (16, 3), (17, 3)\}$. Then we will have the edit distance for g_1, g_2 and g_3 being 1, 0 and 0, respectively, and thus $\text{emError}(W, W') = \frac{1}{3} \cdot (\frac{1}{2} + \frac{0}{3} + \frac{0}{2}) = 0.166$. If the local search algorithm returns a perfect mapping path, the error is zero. Note that the error may exceed 100% if the mapping is way off.

4.2.2 Normalized signal distance of a local search algorithm

Suppose the reference genome is unknown or not accurate (e.g., in many real-world applications such as SNPs or methylation detection), it is difficult to obtain the global mapping. In this case, we may use the normalized signal distance to evaluate the quality of the mapping path W generated by a local search algorithm, which is defined as:

$$\text{nDist}(W) = \frac{\sum_{n=1}^{\text{L}(W)} c(w_{ni}, w_{nj})}{\text{L}(W)}, \tag{5}$$

[1]The nanopore raw signal data were downloaded from nanopore-wgs-consortiumhttp://s3.amazonaws.com/nanopore-human-wgs/rel3-fast5-chr21.part03.tar

[2]https://drive.google.com/drive/folders/1LuOxg9qE1l9AuDcfyUz9aF10X4cgmX5t?usp=sharing

where L(W) is the length of the mapping path W, and $c(w_{ni}, w_{nj})$ is the absolute (or, Z-score) difference of the nth aligned element between the two signal points x_i (the nanopore raw signal) and y_j (the expected signal from the k-mer pore model). Different from emError(W, W'), here nDist(W) is defined over one mapping path W only, instead of over two mapping paths W and W'.

4.3 Performance

4.3.1 Visualization of a detailed example

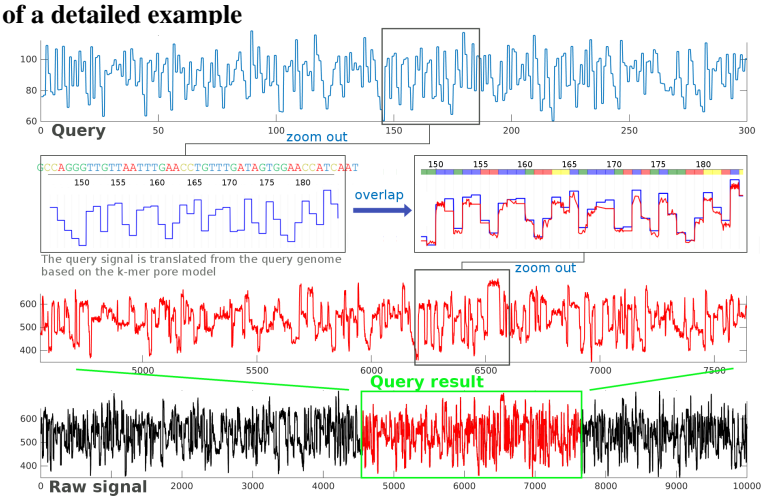

Figure 4: A demonstration of the raw signal similarity search and mapping by our algorithm. Here, the top signal (blue) is a query signal with 300 length, and the bottom signal (black) is a nanopore raw signal with 10,000 length. The zoom-out subfigure locating at [145:185] shows how the query signal corresponds to the $\{A, C, G, T\}$ nucleotide acids. The red signal that locates at [4553:7641] on the raw signal is the query result. By further selecting the segment [6197:6594] on the raw signal and overlapping it with the segment [145:185] on the query signal, a superimposed image with high degree of overlapping is produced.

To demonstrate the effectiveness of our algorithms in discovering the corresponding subsequences in the raw signal sequence, we give one example in Fig. 4 to show the detailed steps of local genome-to-signal search.

Here, a short region of the DNA sequence with 300bp length is selected as the query sequence and a raw signal sequence with 100,000 time points is served as the signal database (the black signal depicted on the bottom of Fig. 4). Both the DNA sequence and the raw signal sequence are selected from the Human21 dataset.

Below are the four steps of the genome-to-signal search procedure:

A) The query sequence is translated into a query signal sequence based on the 6-mer pore model provided by Nanopore Technologies (the blue signal sequence depicted on the top of Fig. 4).

B) Run the DSDTWnano algorithm to obtain the detailed region ([4553:7641]) on the raw signal sequence that has the highest similarity with the query signal sequence (the red signal region depicted on the bottom of Fig. 4). This query operation takes 29 ms and results in a normalized signal distance of 0.1556 between the query signal sequence and the red region of the raw signal sequence. Typically, a normalized signal distance ranging from 0~0.20 indicates a good match.

C) By comparing the zoomed out regions of the query and raw sequences, we can find that these two signals are very similar to each other. However, it should be noted that the query result in the raw signal sequence is about 9× longer than the query signal, which is the typical difference in the sampling speed in nanopore sequencing. Nevertheless, our algorithm still produced accurate mapping.

D) By further selecting the segment [145:185] on the query sequence and the segment [6197:6594] on the raw sequence, we may align and visualize them according to the mapping path produced by our algorithm.

4.3.2 Accuracy analysis

We evaluate the performance of DSDTWnano and cwSDTWnano using the subset of the Human21 dataset and the Lambda phage dataset. In doing so, we randomly select a segment with length l as the query sequence and then run the two algorithms on the corresponding raw signals to find out its maximal response mapping. Finally, we compare the query results of DSDTWnano and cwSDTWnano with the global mapping by the edit mapping error.

We first run an experiment of DSDTWnano and cwSDTWnano (with parameter $K = 3$ and $L = 128$) with the mapping boundary $r = 50$ and the query length $l = 1000$. As shown in Fig. 5(A), the distribution of the edit

mapping error of cwSDTWnano is very similar to that of DSDTWnano, and the majority of the error ranges between 0 and 0.01. Fig. 5(B) shows the scatter plot between the edit mapping error of DSDTWnano (x-axis) and that of cwSDTWnano (y-axis), which indicates that most of them are the same (on the diagonal of the scatter map). The outliers of cwSDTWnano may be caused by the coarsening in the multi-scale analysis.

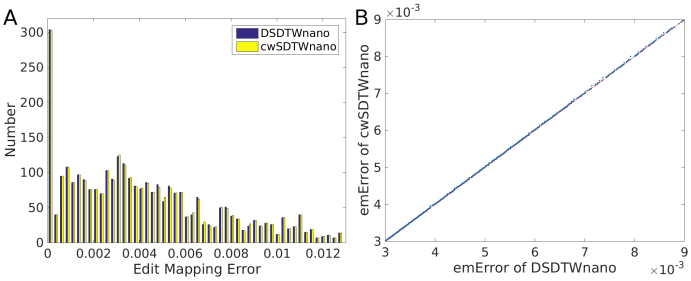

Figure 5: Statistics of the edit mapping error of our algorithms on the Human21 database, where for both DSDTWnano and cwSDTWnano the mapping boundary $r = 50$ and query length $l = 1000$. (A) Distribution of the edit mapping error of DSDTWnano (in yellow) and cwSDTWnano (in blue). (B) Scatter plot between the edit mapping error of the DSDTWnano (x-axis) and that of cwSDTWnano (y-axis).

We then challenge both algorithms with different lengths l of the query sequence and different radius r of the mapping path boundary. The average edit mapping error of the query results by DSDTWnano and cwSDTWnano on the Human21 database and the Lambda phage database are summarized in Tables 1 and 2, respectively. We can find that for queries with different lengths, (i) DSDTWnano almost always outputs a query result within 0.01 edit mapping error, and no larger than 0.006 for most of the cases; (ii) the edit mapping error of cwSDTWnano can also be controlled around 0.006 if a suitable r is selected ($r = 50$ for $l \leqslant 2000$ and $r = 70$ for $l \leqslant 4000$). This is normal as the performance of cwSDTWnano depends on the mapping boundary r which is required for the coarsening of the input signals. As a result, although human and lambda phage are two completely different species, from the little difference between the two tables, we know that the performance of our methods is stable and consistent over different species.

Table 1: The average edit mapping error on the Human21 database

Edit Mapping Error		$l = 600$	$l = 1000$	$l = 2000$	$l = 3000$	$l = 4000$
DSDTW nano	$r = 30$	0.003992	0.004477	0.005322	0.005602	0.005575
	$r = 50$	0.004131	0.004440	0.005213	0.005361	0.005326
	$r = 70$	0.004092	0.004533	0.005147	0.005368	0.005158
cwSDTW nano	$r = 30$	0.004444	0.007988	0.012675	0.019269	0.030104
	$r = 50$	0.004183	0.004651	0.005647	0.005867	0.006013
	$r = 70$	0.004100	0.004598	0.005308	0.005504	0.005395

Table 2: The average edit mapping error on the Lambda phage database

Edit Mapping Error		$l = 600$	$l = 1000$	$l = 2000$	$l = 3000$	$l = 4000$
DSDTW nano	$r = 30$	0.003813	0.003940	0.004631	0.004898	0.004686
	$r = 50$	0.003544	0.004059	0.004527	0.004667	0.004583
	$r = 70$	0.003763	0.003933	0.004347	0.004453	0.004265
cwSDTW nano	$r = 30$	0.004674	0.006521	0.014902	0.035661	0.052850
	$r = 50$	0.003791	0.004549	0.005384	0.005728	0.005973
	$r = 70$	0.003791	0.004294	0.004673	0.004758	0.004667

Because cwSDTWnano has two extra parameters K and L to define the seed number and the seed length, we further analyze the parameter sensitivity. Table 3 summarizes the average edit mapping error on the Human21 database for cwSDTWnano with different seed number K and seed length L (here the search radius r is set to 50). From Table 3 we can find that the seed length has an influence on the quality of the result but the number of seed does not have. Also, the edit mapping error demonstrated in Table 3 indicates that cwSDTWnano is robust for $K \geqslant 3$ and $L \geqslant 128$. Thus a very short seed may cause false dismissals, whereas a seed length of 128 can ensure the correctness of cwSDTWnano for almost all the query, and a seed length of 192 is sufficient for a dataset with raw signals with reasonably good quality.

Table 3: The average edit mapping error of query results on the Human21 database for cwSDTWnano with different configurations

Edit Mapping Error	$l = 600$	$l = 1000$	$l = 2000$	$l = 3000$	$l = 4000$
$K = 3, L = 128$	0.004183	0.004651	0.005647	0.005867	0.006013
$K = 4, L = 128$	0.004155	0.004819	0.005569	0.006062	0.005725
$K = 5, L = 128$	0.004202	0.004643	0.005549	0.005785	0.005832
$K = 3, L = 192$	0.004444	0.004626	0.005355	0.006047	0.005624
$K = 4, L = 192$	0.004223	0.004745	0.005398	0.005949	0.005759
$K = 5, L = 192$	0.004177	0.004950	0.005322	0.006114	0.005730

As the edit mapping error directly reflects how correct the signals aligned upon the nucleotide positions, it can indicate the necessary signal coverage (an analogy to the read coverage) to ensure the success of the DNA mutation analysis, such as SNP detection. A small edit mapping error indicates that there is little theoretical error introduced by our algorithms. Specifically, learning from the <1% edit mapping error produced here, we can infer that a 10× ∼ 20× signal coverage is enough for a mutation analysis if the raw signals are (i) with relative high-quality (∼90% accuracy), and (ii) searched and mapped by our algorithms.

4.3.3 Runtime analysis

For a database with a number of raw signals, the running time for a query task is also very important. Generally, the runtime of DSDTWnano is about 450 ms and that of cwSDTWnano is about 200 ms for a query sequence with 1000bp in length on a 100,000 time points raw signal sequence. When the query length grows, the runtime may increase considerably if there are hundreds or thousands of raw signals. Under this condition, cwSDTWnano is

prepared because it can accelerate the query process dramatically by a multi-scale strategy. In this subsection, the runtime for both DSDTWnano and cwSDTWnano is investigated.

Fig. 6 demonstrates the runtime of our algorithms. All the execution time is collected on a Fedora25 system with 128Gb memory and two E5-2667v4 (3.2 GHz) processors. From Fig. 6(A) we can find that DSDTWnano has a much higher execution time compared with cwSDTWnano when the query length increases, whereas cwSDTWnano keeps a low computational cost. Especially, the runtime of cwSDTWnano is always smaller than 900 ms even searching a 6000bp-long query on a raw signal with 2×10^5 time points. From Fig. 6(B) we can find that the runtime of cwSDTWnano will not exceed 1500 ms when searching a 1000bp-long query on a raw signal with 1×10^6 time points.

In summary, though the introduction of multi-scale coarsening analysis in cwSDTWnano causes a degeneration in mapping accuracy (Tables 1 and 2), the loss of accuracy is acceptable when considering the benefit to the runtime. In practice, we will (i) run DSDTWnano if the query length is short, and (ii) run cwSDTWnano if the query is long.

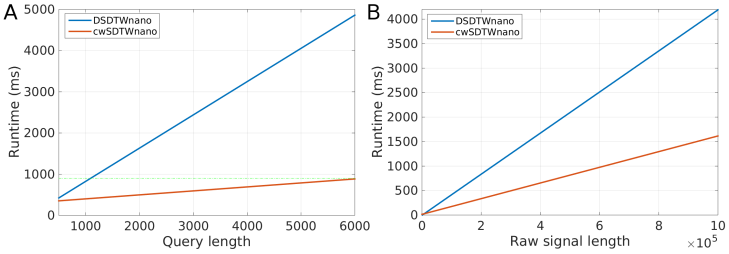

Figure 6: The runtime of our algorithms with different query lengths and raw signal lengths ($r = 50$, $K = 3$ and $L = 128$). (A) The runtime of DSDTWnano and cwSDTWnano on a 2×10^5-long raw signal sequence when the length of the query changes; (B) The runtime of DSDTWnano and cwSDTWnano for a 1000bp-long query when the length of raw signals changes.

4.3.4 Discrimination of hit and non-hit signals

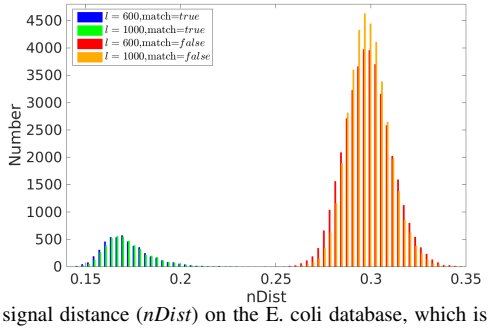

Figure 7: The histogram of the normalized signal distance (*nDist*) on the E. coli database, which is generated from 200 query sequences whose length is 600bp-long or 1000bp-long. Each query sequence has 200 $non-hit$ raw signals (denoted as the $false$ set) and about 25 *hit* raw signals (denoted as the *true* set).

A fundamental task in nanopore sequencing is that, given a query sequence and a raw signal database, whether we can find a set of signal segments (subsequences of raw signals) that are similar to the query. Here we use the concept hit (non-hit) to indicate the raw signals that contain (do not contain) a signal segment with high-similarity to the query. This is necessary because in some applications, the task is to find some non-standard events (e.g., SNPs or DNA methylations), in which multiple numbers of hit signals are required for the reliable detection of these events. However, it is not easy to implement a scoring function to discriminate hit and non-hit signals, especially when the reference genome is unknown or not accurate. In this case, the global mapping is unknown and whether a raw signal sequence contains the query sequence is also unknown. Here, the normalized signal distance (*nDist*) is proposed to filter the non-hit signals, without the need of the reference genome.

We use the E. coli database to test the performance of *nDist* with DSDTWnano to identify hit and non-hit signals. To construct the benchmark, we randomly select 600bp-long and 1000bp-long subsequences from the E. coli genome as the query sequences, each with 200 samples. For each raw signal sequence in the E. coli database, as its corresponding reference sequence is known, we are able to get the true label of each sequence. Since the sequencing coverage of the E. coli dataset is around 20, we use all the hit signals as the *true* set, and randomly sample 200 non-hit as the $false$ set. For each pair of the query sequence and the raw signals in either the *true* or $fasle$ set, we run DSDTWnano to obtain the local mapping path and the corresponding *nDist* score.

As shown in Fig. 7, we observe that (i) there are two well-separated distributions of *nDist*, where most of the left (right) belongs to the *true* ($fasle$) set; (ii) *nDist* from the query sequence with different lengths reside in the same distribution. Thus, *nDist* can distinguish the *true* and $false$ set no matter the query sequence is 600bp-long or

1000bp-long. It is obvious that there is a clear boundary around $nDist = 0.2$ that could separate the *true* and $false$ sets. Therefore, to obtain hit signals for the query sequence, we may set the threshold for $nDist$ to 0.2 during the database search. This result ensures the accurate identification of similar subsequences in raw signals to the query, which is essential to the detection of SNPs in the signal level.

4.4 Case study

Currently, the identification of SNPs is mainly achieved by resequencing approach (i.e., searching for differences between aligned reads and the reference genome) or assembly approach (i.e., *de novo* assembling consensus read sequences against a reference genome) [20]. Recently, a few works explored the capability of nanopore sequencing to identify SNPs [21], which shows that to reach a high detection rate (such as > 95%), more than 60× sequencing coverage is needed[22].

Here, a case study is presented to demonstrate how we can identify and visualize SNPs based on the nanopore raw signals by our algorithm. The experiment is carried out on the E. coli dataset with a relatively low coverage (20×). In doing so, we first generate a synthetic reference genome by randomly substituting 100 bases on the reference genome. Then we extract a 2000bp-long sequence on the synthetic genome that might contain a SNP and search it against the nanopore signal database by cwSDTWnano, to locate the candidate raw signal segments and positions that might contain a SNP. After that, for each candidate position, four mutated sequences each with that position being $\{A, C, G, T\}$, respectively, are used as the query to search against the signal database. Finally, the mutation with the expected signal closest to the observed signals in the database is chosen as the detected SNP at the candidate position (more details are given in Section S3).

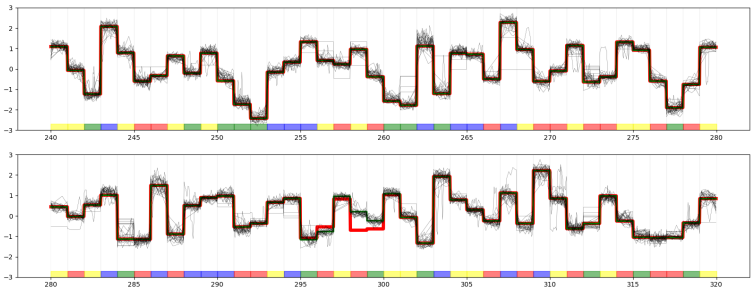

Figure 8: Illustration of SNP detection by cwSDTWnano, which is a segmentation from a set of aligned raw signals. Here, the nucleotide A, C, G and T on the reference sequence (query) are labeled in red, yellow, green and blue , respectively. The aligned nanopore signals are shown in the black squiggle curves and the red (green) curves indicate the 6-mer pore model for the reference (mutated) sequence centered at the candidate of SNP position.

An example of SNP identification is shown in Fig. 8, which is a region of aligned raw signals (a full mapping can be found in Section S3). Here the red (green) curves indicate the 6-mer pore model for the reference (mutated) sequence centered at the candidate SNP position. The aligned nanopore signals are shown in black. There is a clear difference of the pore model at the SNP position, which indicates a strong evidence. Experiments on the E. coli dataset demonstrate that accurate SNP detection (around 95%) can be achieved by a low coverage (i.e., 20×). The success of our algorithms lies in two folds: (i) the normalized signal distance measurement filters out low-quality signals, and (ii) the signal-level operation guarantees a better searching and mapping.

5 Discussion and conclusion

We proposed two subsequence search algorithms for the genome-to-signal similarity search and mapping. The proposed algorithms are based on the idea of DTW and directly operate on the nanopore raw signals. Furthermore, a novel error measurement is proposed to specify the mapping accuracy between a genomic sequence and a raw signal sequence. Comprehensive experiments on three real-world nanopore datasets demonstrate the proposed algorithm is able to produce accurate and effective subsequence search, and detect SNPs with a low sequencing coverage.

Our proposed algorithms could also be extended and applied to detect other single nucleotide variants (SNV), such as small insertions and deletions (InDels), as well as DNA modifications such as C^5-methylcytosine (5-mC) and N^6-methyladenine (6-mA). Reports have shown that these events are very challenging for most of the current tools to detect, especially at a low sequencing coverage or low-quality raw signals.

Acknowledgements

The authors thank Minh Duc Cao, Lachlan J.M. Coin, Louise Roddam and Tania Duarte for providing the nanopore sequencing data. This work was supported by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR) under Awards No. FCC/1/1976-04, URF/1/2601-01, URF/1/3007-01, URF/1/3412-01, and URF/1/3450-01.

References

[1] M. Jain, H. E. Olsen, B. Paten, and M. Akeson, "The oxford nanopore minion: delivery of nanopore sequencing to the genomics community," *Genome biology*, vol. 17, no. 1, p. 239, 2016.

[2] D. Deamer, M. Akeson, and D. Branton, "Three decades of nanopore sequencing," *Nature biotechnology*, vol. 34, no. 5, p. 518, 2016.

[3] M. C. Stancu, M. J. Roosmalen, I. Renkens, M. M. Nieboer, S. Middelkamp, J. Ligt, G. Pregno, D. Giachino, G. Mandrile, J. E. Valle-Inclan, *et al.*, "Mapping and phasing of structural variation in patient genomes using nanopore sequencing," *Nature Communications*, vol. 8, no. 1, p. 1326, 2017.

[4] F. Guo, D. Wang, and L. Wang, "Progressive approach for snp calling and haplotype assembly using single molecular sequencing data," *Bioinformatics*, vol. 34, no. 12, pp. 2012–2018, 2018.

[5] F. J. Rang, W. P. Kloosterman, and J. de Ridder, "From squiggle to basepair: computational approaches for improving nanopore sequencing read accuracy," *Genome biology*, vol. 19, no. 1, p. 90, 2018.

[6] A. C. Rand, M. Jain, J. M. Eizenga, A. Musselman-Brown, H. E. Olsen, M. Akeson, and B. Paten, "Mapping dna methylation with high-throughput nanopore sequencing," *nature methods*, vol. 14, no. 4, pp. 411–413, 2017.

[7] M. H. Stoiber, J. Quick, R. Egan, J. E. Lee, S. E. Celniker, R. Neely, N. Loman, L. Pennacchio, and J. B. Brown, "De novo identification of dna modifications enabled by genome-guided nanopore signal processing," *bioRxiv*, p. 094672, 2016.

[8] R. Han, Y. Li, X. Gao, and S. Wang, "An accurate and rapid continuous wavelet dynamic time warping algorithm for end-to-end mapping in ultra-long nanopore sequencing," *Bioinformatics*, vol. 34, no. 17, pp. i722–i731, 2018.

[9] K. Berlin, S. Koren, C.-S. Chin, J. P. Drake, J. M. Landolin, and A. M. Phillippy, "Assembling large genomes with single-molecule sequencing and locality-sensitive hashing," *Nature biotechnology*, vol. 33, no. 6, p. 623, 2015.

[10] C. Jain, S. Koren, A. Dilthey, A. M. Phillippy, and S. Aluru, "A fast adaptive algorithm for computing whole-genome homology maps," *Bioinformatics*, vol. 34, no. 17, pp. i748–i756, 2018.

[11] H. Li, "Minimap2: pairwise alignment for nucleotide sequences," *Bioinformatics*, vol. 1, p. 7, 2018.

[12] Y. Sakurai, C. Faloutsos, and M. Yamamuro, "Stream monitoring under the time warping distance," in *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pp. 1046–1055, IEEE, 2007.

[13] T. Saramaki and R. Bregovic, "Multirate systems and filterbanks," in *Multirate systems: design and applications*, pp. 27–85, IGI Global, 2002.

[14] C. A. Ratanamahatana and E. Keogh, "Three myths about dynamic time warping data mining," in *Proceedings of the 2005 SIAM International Conference on Data Mining*, pp. 506–510, SIAM, 2005.

[15] S. Salvador and P. Chan, "FastDTW: Toward accurate dynamic time warping in linear time and space," *Intelligent Data Analysis*, vol. 11, no. 5, pp. 561–580, 2007.

[16] M. Müller, H. Mattes, and F. Kurth, "An efficient multiscale approach to audio synchronization.," in *ISMIR*, pp. 192–197, 2006.

[17] T. Prätzlich, J. Driedger, and M. Müller, "Memory-restricted multiscale dynamic time warping," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pp. 569–573, IEEE, 2016.

[18] H. Skutkova, M. Vitek, K. Sedlar, and I. Provaznik, "Progressive alignment of genomic signals by multiple dynamic time warping," *Journal of theoretical biology*, vol. 385, pp. 20–30, 2015.

[19] M. Jain, S. Koren, K. H. Miga, J. Quick, A. C. Rand, T. A. Sasani, J. R. Tyson, A. D. Beggs, A. T. Dilthey, I. T. Fiddes, *et al.*, "Nanopore sequencing and assembly of a human genome with ultra-long reads," *Nature biotechnology*, vol. 36, no. 4, p. 338, 2018.

[20] A. Magi, R. Semeraro, A. Mingrino, B. Giusti, and R. D'aurizio, "Nanopore sequencing data analysis: state of the art, applications and challenges," *Briefings in bioinformatics*, 2017.

[21] J. Quick, N. J. Loman, S. Duraffour, J. T. Simpson, E. Severi, L. Cowley, J. A. Bore, R. Koundouno, G. Dudas, A. Mikhail, *et al.*, "Real-time, portable genome sequencing for ebola surveillance," *Nature*, vol. 530, no. 7589, p. 228, 2016.

[22] M. Jain, I. T. Fiddes, K. H. Miga, H. E. Olsen, B. Paten, and M. Akeson, "Improved data analysis for the minion nanopore sequencer," *Nature methods*, vol. 12, no. 4, pp. 351–356, 2015.