Genome Biology

SOFTWARE

Open Access

TDFPS-Designer: an efficient toolkit for barcode design and selection in nanopore sequencing

Junhai Qi[1†], Zhengyi Li[1†], Yao-zhong Zhang[2], Guojun Li[1*], Xin Gao[3*] and Renmin Han[1*]

†Junhai Qi and Zhengyi Li are joint first author.

*Correspondence:
gjli@sdu.edu.cn;
xin.gao@kaust.edu.sa;
hanrenmin@sdu.edu.cn

[1] Research Center for Mathematics and Interdisciplinary Sciences, Shandong University, Qingdao 266237, China
[2] Division of Health Medical Intelligence, Human Genome Center, The Institute of Medical Science, The University of Tokyo, Minato-ku, Tokyo 108-8639, Japan
[3] Computational Bioscience Research Center (CBRC), Computer, Electrical and Mathematical Sciences and Engineering (CEMSE) Division, King Abdullah University of Science and Technology (KAUST), Thuwal, Makkah 23955, Saudi Arabia

Abstract

Oxford Nanopore Technologies (ONT) offers ultrahigh-throughput multi-sample sequencing but only provides barcode kits that enable up to 96-sample multiplexing. We present TDFPS-Designer, a new toolkit for nanopore sequencing barcode design, which creates significantly more barcodes: 137 with a length of 20 base pairs, 410 at 24 bp, and 1779 at 30 bp, far surpassing ONT's offerings. It includes GPU-based acceleration for ultra-fast demultiplexing and designs robust barcodes suitable for high-error ONT data. TDFPS-Designer outperforms current methods, improving the demultiplexing recall rate by 20% relative to Guppy, without a reduction in precision.

Background

Recently, single-molecule sequencing based on ONT has emerged, offering freedom from long reads, point-of-care, and polymerase chain reactions (PCRs). Specifically, ONT has been widely applied in various research fields, including genome assembly [1–6], transcriptome assembly [7–9], methylation research [10–12], and mutation identification [13–15]. To efficiently utilize sequencing capacity and reduce sequencing costs, multiple DNA/RNA samples can be integrated with unique barcodes and sequenced simultaneously on a flow cell [16]. After sequencing, demultiplexing is necessary to classify the sequences according to their corresponding barcodes. To address the demultiplexing problem, several methods have been introduced in recent years, such as DeepBinner [17] and DeePlexiCon [18]. These methods utilize convolutional neural networks (CNNs) to directly process the native nanopore signals for demultiplexing, improving upon traditional sequence-based tools like Porechop. However, they do not explore which barcodes are most conducive to effective demultiplexing. Currently, ONT provides a barcode kit (EXP-PBC096) that supports the simultaneous sequencing of up to 96 samples. As the number of samples increases, an additional strategy is needed for large-capacity multiple-sample sequencing [19]. A direct solution is to design specified barcodes for accurate and large-capacity sample demultiplexing.

Barcode design can be viewed as an error-correcting code design problem, and related theories have been developed since the 1970s [20, 21]. To address the needs of high-throughput next-generation sequencing, Hamming codes and Reed-Solomon code barcodes have been introduced into DNA barcode design. Hamady et al. [22] developed a new set of barcodes based on error-correcting codes. Zorita et al. [23] described an exact algorithm to determine which pairs of sequences lie within a given Levenshtein distance. Hawkins et al. [24] presented and experimentally validated filled/truncated right-end edit (FREE) barcodes, which corrected substitution, insertion, and deletion errors for next-generation sequencing. Although numerous barcode schemes have been proposed, these schemes are designed on the prerequisite that the sequencing error rate is very low (less than 1%), which means that these schemes are likely not applicable to third-generation sequencing data with higher sequencing error (∼6–15% [5]). In the context of nanopore sequencing, [25] utilized an evolutionary model to design 96 “Molbit barcodes” that ensured dissimilarity in nanopore electrical signals. A specially trained convolutional neural network (CNN) was employed to accurately demultiplex these barcodes. However, this approach did not produce a kit with a larger capacity than the ONT barcode kit, limiting its utility for multi-sample sequencing involving a greater number of samples.

Barcode design must observe two key principles, i.e., large barcode capacity and high sequence difference. For ONT sequence data, the measure of sequence difference could be based on either the raw current signal or base-called nucleotides. Edit distance [26] can effectively measure the similarity between two DNA sequences. However, relying solely on edit distance for demultiplexing can result in the loss of a significant amount of useful data. To further improve edit distance, some approaches take into account the quality score of each base obtained after sequencing. This score is highly correlated with the probability that the base has been correctly sequenced. Some quality-aware probabilistic methods that account for these quality scores have been applied to sequence error correction [27] and demultiplexing problems [28] in next-generation sequencing (NGS). Many alignment-free similarity measures have also been proposed [29–32]. In contrast, signal-based approaches [17, 18, 33, 34] have been widely utilized in direct nanopore sequence analysis, most of which are based on the dynamic time warping (DTW) algorithm to measure the signal difference [35–37]. Just as probabilistic methods account for substitution errors in NGS, DTW addresses inherent error profiles by directly comparing raw nanopore signals.

In this study, we propose a Designer for a barcode kit that employs a well-defined Threshold to reduce the sampling space of the DTW-based Farthest Point Sampling algorithm (TDFPS-Designer) for accurate barcoded sample demultiplexing in nanopore sequencing. TDFPS-Designer selects barcodes within a given sequence space by the farthest point sampling algorithm, directly based on the comparison of nanopore signals. Additionally, a DTW distance-based demultiplexing strategy is designed to ensure accurate sample label assignment. Three barcode kits with different barcode lengths were designed by TDFPS-Designer. Experiments demonstrated that TDFPS-Designer is capable of designing barcode sets with $\geq$ 99% demultiplexing accuracy, superior to the randomly selected barcodes and ONT official strategy. Specifically, there is almost no "collision" during the demultiplexing of TDFPS-Designer's barcode set. When demultiplexing large-capacity samples with high sequencing error rates, the demultiplexing recall of TDFPS-Designer's barcode kit is approximately 20% higher than that of current official ONT tools, which provides an alternative for the demultiplexing of barcode kits with high sequencing error.

Results

Algorithms overview

TDFPS-Designer selects barcode candidates from a specified set of an entire k-mer space or user-defined sequences. The workflow of TDFPS-Designer is illustrated in Fig. 1. The sampling space is first reduced to a subset of sufficiently distinct sequences, such that the DTW distance between any two sequences in the subset is greater than the threshold *r* (Fig. 1a). Here, we begin by randomly selecting a sequence, and the relationship between the randomly selected sequence and the final set of barcodes is explored in detail in Additional File 1: S1. The demultiplexing strategy of TDFPS-Designer is depicted in Fig. 1b, where demultiplexing is performed directly from the DTW distance

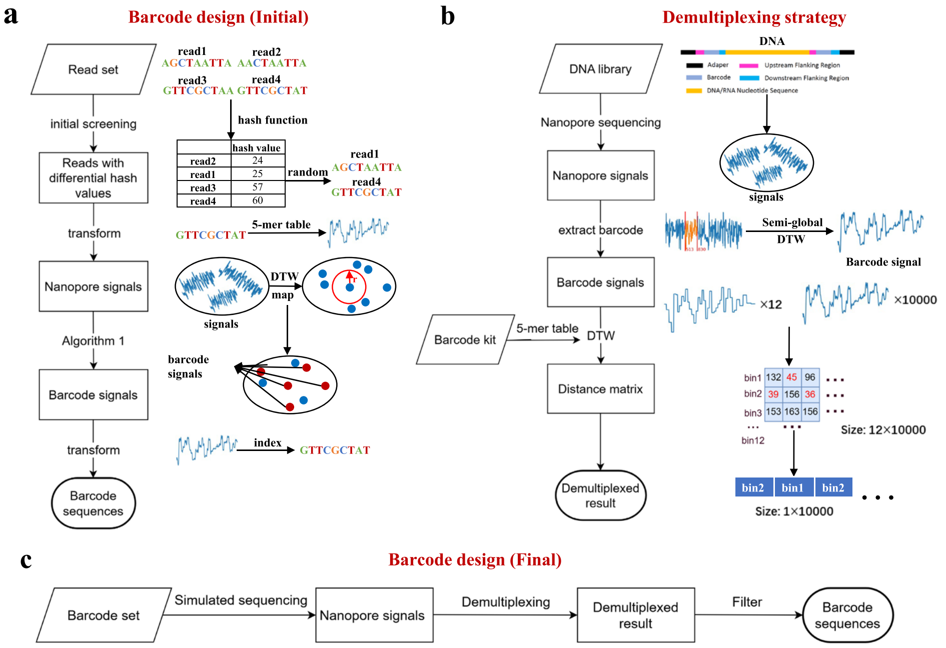


Fig. 1 The workflow of TDFPS-Designer. **a** Barcode design strategy (initial). Given a k-mer space or a user-defined sequence set, the hash value (Eq. (2)) of each read is calculated, and the sequences are sorted according to the hash value in ascending order. Next, a subset of sequences is selected evenly from the sorted sequence items. The hash values of the selected sequences should have significant distinctness. These selected sequences are then converted into simulated nanopore current signals. Algorithm 1 is used to select the exact signals as barcode signals, ensuring that the DTW distances between these signals are relatively large and greater than a given threshold *r*. **b** Demultiplexing strategy. To demultiplex nanopore signals, the barcode regions of reads are identified and compared to the current signals translated from the standard barcode references (items of the barcode sets). Afterwards, the DTW distance matrices between the sequenced barcode signals and the standard barcode current signals are calculated, based on which a top-k selection is carried out to determine the demultiplexing results. **c** Barcode design strategy (final). Based on the initially designed barcodes, the demultiplexing process is simulated. Barcodes with poor demultiplexing performance are filtered out, resulting in the final kit of barcodes

matrix. To further enhance the robustness of the initially designed barcodes, our method can automatically simulate the demultiplexing process and filter out barcodes with poor demultiplexing performance, resulting in the final set of designed barcodes (Fig. 1c). The use of TDFPS-Designer is described comprehensively in Additional File 1: S2.

Benchmark datasets

All simulated datasets were generated by *DeepSimulator1.5*, *squigulator* [38], *Badread* [39], and our own multisample sequencing simulator. The simulated electrical signals could achieve ∼88–92% base-calling accuracy (Additional File 1: S3), while the real-world electrical signals based on MinION R9.4 can achieve ∼85–94% sequencing accuracy [5], which indicates that the difference between the simulated electrical current signals and the real electrical current signals is negligible. In addition, we carefully studied the construction process of the ONT multisample sequencing library to ensure maximum consistency between the simulated data and the real data. We also employed *Badread* to produce sequences with different sequencing error rates. This allowed us to examine the impact of sequencing error rates on demultiplexing, which influences both the capacity of the resulting barcode kit and the selection of demultiplexing strategies.

A few small datasets were generated for the initial evaluation of our barcode design strategy (Additional File 1: S4.1). On the other hand, we thoroughly investigated the library preparation process of ONT and integrated it into our data simulator. The detailed use of the simulator and the introduction of parameters are given in Additional File 1: S4.2. Based on our data simulator, we generated different types of datasets, and the details of these multisample sequencing datasets are shown in Table 1. The detailed data generation process can be found in Additional File 1: S4.3.

Table 1 Datasets for evaluating all methods

Dataset	Sample	Type of barcode kit	Size of barcode kit	Barcode length (bp)	Number of reads
S-ET_ONT12	ETEC	ONT_EXP-NBD104	12	24	12,000
S-ET_ONT24	STEC	ONT_SQK-16S024	24	24	24,000
S-ET_ONT96	ETEC, STEC, and HS	ONT_EXP-PBC096	96	24	96,000
M-ESH_TD795	ETEC, STEC, and HS	Initially designed by TDFPS-Designer	795	20	795,000
M-ESH_TD1093	ETEC, STEC, and HS	Initially designed by TDFPS-Designer	1093	24	1,093,000
M-ESH_TD2120	ETEC, STEC, and HS	Initially designed by TDFPS-Designer	2120	30	2,120,000
L-ESH_TD137	ETEC, STEC, and HS	Finally designed by TDFPS-Designer	137	20	691,850
L-ESH_TD410	ETEC, STEC, and HS	Finally designed by TDFPS-Designer	410	24	2,070,500
L-ESH_TD1779	ETEC, STEC, and HS	Finally designed by TDFPS-Designer	1779	30	8,983,950

Here, the full name of ETEC is *Enterotoxigenic Escherichia coli*, the full name of STEC is *Shiga toxin-producing Escherichia coli*, and the full name of HC is *Historical Shigella*. For "ONT_EXP-NBD104," "ONT" implies that it is designed by ONT, and "EXP-NBD10" is its kit name. "ONT_EXP-NBD104" and "ONT_SQK-16S024" imply the same meaning as "ONT_EXP-NBD104." Specifically, L-ESH_TD137, L-ESH_TD410, and L-ESH_TD1779 include 1% negative samples that were not successfully barcoded. These negative samples serve as a "noise class"

Evaluation metrics

The goal of our approach is to design enough barcodes with different lengths that can be easily demultiplexed. For this purpose, we evaluate the demultiplexing performance of different demultiplexing algorithms (Guppy and our method) on our designed barcode kits using precision, recall, average accuracy and F1-score, which reflect whether the barcodes we designed can be easily demultiplexed. Precision measures how many instances are indeed positive given that the model predicted some instances to be positive. In simple terms, precision reflects the credibility of the model's prediction of positive samples. Recall, also known as the true positive rate or sensitivity, quantifies the ability of a model to capture all positive examples from a dataset. Each barcode corresponds to a precision (recall, F1-Score). For example, after demultiplexing, assuming that the barcode label of sequence in $\{read_1, read_2..., read_n\}$ is *barcode*, the set of sequences that actually carry this barcode is B, then the formula to calculate recall for this barcode is $\frac{|\{read_1,read_2...,read_n\}\cap B|}{|B|}$. Once we obtain all the indicators corresponding to all barcodes, the average of all accuracy rates is recorded as the average accuracy, the minimum precision (recall, F1-score) is recorded as the minimum precision (recall, F1-score), and the second minimum precision (recall, F1-score) is recorded as minimum-2 precision (recall, F1-score). When working with numerous barcodes, a high average accuracy in demultiplexing results does not necessarily mean a consistently high accuracy across all barcodes. There might be instances where the algorithm performs well for most barcodes but poorly for specific ones. Relying solely on average accuracy might not offer a complete assessment of demultiplexing effectiveness. By considering minimum/minimum-2 precision (recall, F1-score) alongside average accuracy, we can gain a more comprehensive understanding of the algorithm's performance. All metrics are calculated using the "sklearn" Python package.

Experimental environment

All the experiments were run on an Ubuntu 18.04.6 system with an Intel(R) Xeon(R) Platinum 8260 CPU, 1 Tb memory, and an A100-PCIE-40GB.

TDFPS-Designer can effectively extract the barcode region from the raw nanopore signal to ensure accurate demultiplexing results

We assessed the effectiveness of our barcode extraction strategy by calculating the DTW distance between the extracted barcode signals and the standard barcode signals. To generate experimental data, we obtained 12,000 extracted barcode signals and 1000 randomly intercepted signals, from which we obtained two distance matrices (Fig. 2a). Based on these matrices, we generated two different distance distributions (Fig. 2c). As shown in Fig. 2c (right), the probability that the distance between a signal and the standard barcode signal is less than 110 is very low (~0.0061). In contrast, Fig. 2c (left) shows that 94.35% of the DTW distances between the extracted barcode signals and the standard barcode signals are less than 110, indicating that our extraction strategy is highly effective. In terms of efficiency, by using a single thread, we can extract the barcode regions of approximately 255 sequences in just 1 s.

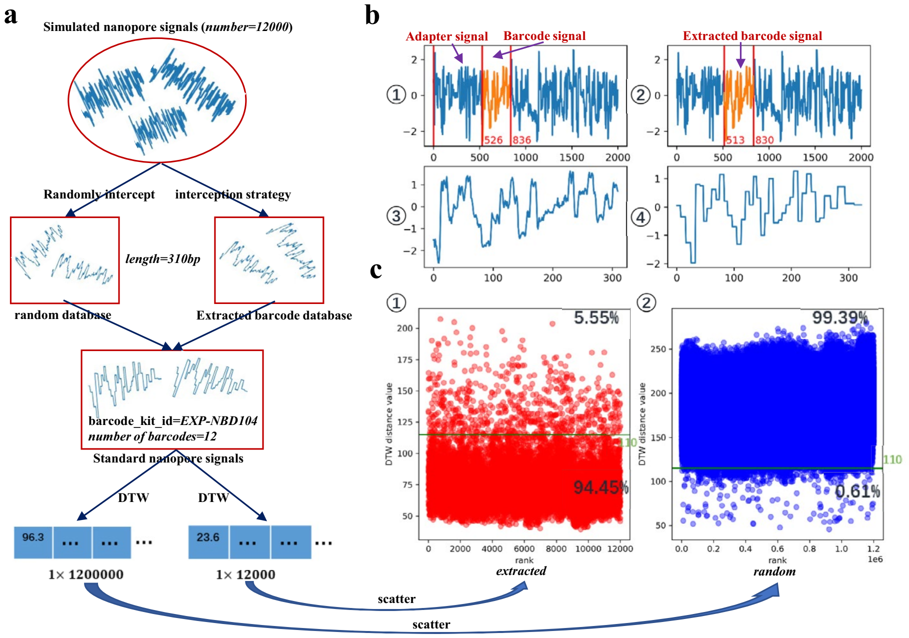


Fig. 2 **a** Based on the barcode kit (EXP-NBD104, 12 barcodes), a total of 12,000 simulated nanopore signals are generated, and each barcode corresponds to 1000 signals. We randomly intercept 1000 signal lengths of 310 in each signal (left) and use the extraction strategy to obtain the barcode signal (length = 310). The distance matrix between these two groups of signals and the standard nanopore signal of the barcode is calculated, and the corresponding distance distribution is obtained. **b** ① The positions of the barcode signal and adapter signal. The red number indicates the position of the barcode signal. ② The position of the barcode signal obtained by the extraction strategy. ③ Nanopore signal image of barcode. ④ Image of standard (or noiseless) barcode nanopore signal. **c** ① Distribution of DTW distances between the extracted barcode signals and the standard barcode signals. The proportion of distance values above 110 is 5.55%. ② Distribution of DTW distances between randomly selected signals and standard barcoded signals. The proportion of distance values above 110 is 99.39%

TDFPS-Designer can design specialized barcodes for different sequencers

ONT offers various sequencers, such as the MinION sequencer and PromethION sequencer, each with different chemistries, such as R9.4 and R10.4. The R9.4 has been widely adopted, demonstrating mature and stable performance, while the R10.4 aims to further enhance sequencing accuracy, and they may generate different nanopore signals (Fig. 3a). For each sequencer, we designed 96 barcodes, each 20 bp in length, matching the capacity of the ONT barcode kit but with shorter lengths. We generated different types of nanopore signals (use *Squigulator*) based on these barcodes (with 100 simulated signals per barcode) and used TDFPS-Designer for demultiplexing. The results showed that these barcodes could be accurately demultiplexed (Fig. 3b), suggesting that our algorithm can customize barcode kits for different sequencers. In the subsequent analysis, we primarily discuss the barcodes designed for the MinION R9.4.

Barcodes designed by TDFPS-Designer are easier to demultiplex than randomly selected barcodes

In biological experiments, barcodes are often randomly selected as short DNA fragments using various methods, such as random nucleic acid synthesis or selection from existing barcode libraries. We evaluated the effectiveness of our barcode design

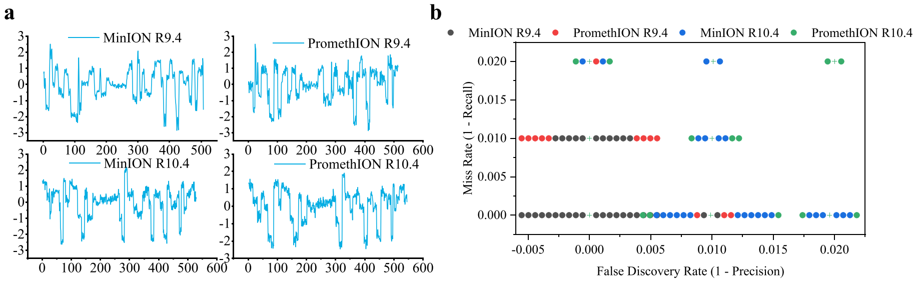


Fig. 3 Performance of TDFPS-Designer for four different sequencers. **a** The electrical signals generated by "GGCGTCTGCTTGGGTGTTTAACCTTTTTTTTTTTAATGTACTTCGTTCAGTTACGTATTGCT" on four sequencers. **b** The demultiplexing performance of TDFPS-Designer on four different barcode kits. These barcode kits, each containing 96 barcodes of 20 bp in length, were specifically designed by TDFPS-Designer for four different sequencers. For better visualization, all overlapping points have been expanded horizontally

strategy based on the accuracy of demultiplexing. We first used both a random strategy and TDFPS-Designer to design 100 barcodes, each 15 bp in length, and evaluated their demultiplexing performance (Fig. 4a). We found that some barcodes generated by the random strategy could not be demultiplexed accurately, with an precision of less than 0.84, which is ∼10% lower than the ones of TDFPS-Designer, and every barcode designed by TDFPS-Designer could be accurately demultiplexed. Additionally, we used both strategies to generate 96 barcodes, each 24 bp in length, and compared them with ONT barcodes (Fig. 4b). The results showed that all three types of barcodes had stable demultiplexing performance, which can be attributed to the large sequence space of the 24 bp barcodes, leading to a very low probability of collisions between randomly generated barcodes (i.e., one barcode being mistakenly demultiplexed as another barcode). To further validate our findings, we designed 500 barcodes, each 24 bp in length, using both the random strategy and TDFPS-Designer, and evaluated their demultiplexing performance (Fig. 4c). The results indicated that barcodes designed by TDFPS-Designer outperformed those generated randomly, suggesting a tendency for collisions between randomly generated barcodes in this case.

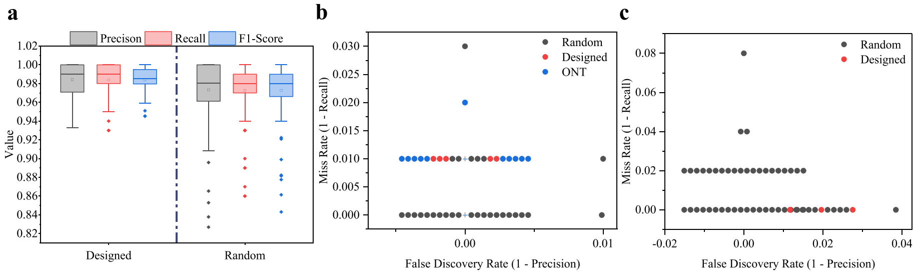


Fig. 4 Performance comparison between random strategy and TDFPS-Designer. **a** Demultiplexing results of 100 15-bp barcodes generated by random strategy and 100 15bp barcodes designed by TDFPS-Designer. "Designed" indicates barcodes designed by TDFPS-Designer. "Random" indicates randomly generated barcodes. **b** Demultiplexing results of 96 24-bp barcodes generated by random strategy, designed by TDFPS-Designer, and ONT. For better visualization, all overlapping points have been expanded horizontally. **c** Demultiplexing results of 500 24bp barcodes generated by random strategy and 500 24-bp barcodes designed by TDFPS-Designer. For better visualization, all overlapping points have been expanded horizontally

TDFPS-Designer can design large-capacity barcode kits with different lengths and ensure their stable demultiplexing

Based on TDFPS-Designer, we designed three final barcode kits, each derived from an initial kit that ensures the difference in DTW distance (see the "Methods" section). TDFPS-Designer provides demultiplexing functionality, and we conducted preliminary tests on the demultiplexing performance of TDFPS-Designer on these kits, comparing it with Guppy (Table 2). We can see that our demultiplexing method and Guppy achieve almost perfect demultiplexing results on the three datasets (S-ET_ONT12, S-ET_ONT24, and S-ET_ONT96) with ONT barcodes. Guppy's demultiplexing method is specially designed for ONT barcodes, so its minimum F1-Score is slightly higher than TDFPS-Designer by 1% to 4%, and the average accuracy is almost the same. In demultiplexing both the initial and final barcode kits, TDFPS-Designer demonstrated higher demultiplexing accuracy, exceeding Guppy by 4% to 9%. Additionally, Guppy classified a large number of reads as unclassified, which is costly. We have constructed a more detailed analysis of this aspect below. On the other hand, by observing the minimum/minimum-2 F1-score, we can see that both Guppy and TDFPS-Designer do not perform well in demultiplexing certain barcodes in the initial kits. This indicates the necessity for TDFPS-Designer to further filter barcodes from the initial kits (see the "Methods" section, Fig. 1c). In the final designed barcode kits, TDFPS-Designer showed nearly perfect demultiplexing performance, with an accuracy greater than 99%, exceeding Guppy by 9%, and with minimum/minimum-2 F1-scores greater than 95%, surpassing Guppy by 8%. These results suggest that TDFPS-Designer can successfully demultiplex all barcodes in the final kits. It is worth noting that Guppy's minimum F1-score was only ~0.17, as it classified a large number of reads as "unclassified" when testing the final

Table 2 Classification performance of demultiplexing tools on benchmark datasets

Dataset	Unclassified		Minimum/minimum-2 F1-score		Average accuracy		Run time (m:s)	
	Guppy6.5.7	TDFPS-Designer	Guppy6.5.7	TDFPS-Designer	Guppy6.5.7	TDFPS-Designer	Guppy6.5.7	TDFPS-Designer
S-ET_ONT12	12	0	0.997/0.998	0.989/0.993	0.999	0.995	0:03	0:06
S-ET_ONT24	66	0	0.995/0.996	0.971/0.978	0.997	0.993	0:09	0:14
S-ET_ONT96	306	0	0.988/0.991	0.964/0.965	0.997	0.994	1:18	1:21
M-ESH_TD795	71,913	34	0.768/0.820	0.819/0.839	0.902	0.967	37:54	18:03
M-ESH_TD1093	97,384	238	0.851/0.860	0.767/0.833	0.909	0.947	94:17	39:54
M-ESH_TD2120	188,586	588	0.874/0.875	0.924/0.934	0.910	0.979	468:03	130:10
L-ESH_TD137	71,833	7083	0.174/0.873	0.953/0.958	0.904	0.994	7:33	5:48
L-ESH_TD410	212,769	21,537	0.175/0.881	0.969/0.975	0.906	0.997	47:46	32:05
L-ESH_TD1779	882,063	95,294	0.183/0.877	0.965/0.996	0.911	0.999	1003:38	741:15

All datasets were generated by *Badread* (R9.4). "Run time" refers to the GPU-accelerated demultiplexing time when barcode regions are extracted, which is obtained by the "time -v" command kits, leading to poor precision in the “noise class” (see Table 1) and resulting in a very low minimum F1-score. In terms of efficiency, when barcode regions in all sequences are extracted, our method is faster than Guppy, which benefits from a well-designed GPU acceleration mechanism [40].

TDFPS-Designer is more robust than Guppy in handling sequencing errors

We evaluated Guppy’s and TDFPS-Designer’s demultiplexing performance on datasets with different sequencing error rates (Fig. 5a). Figure 5b shows Guppy’s demultiplexing performance on three datasets with initial barcode kits. We can see that sequencing errors severely impact the performance of Guppy, with a minimum recall of less than 65% on M-ESH_TD795 (Guppy R9.4), implying that some barcodes were not successfully demultiplexed, and we can see that almost all barcodes are effectively demultiplexed when the sequencing error rate is lower (Guppy R10.4). In addition, we can see from Fig. 5c that both Guppy and TDFPS-Designer exhibit high demultiplexing precision. However, Guppy shows relatively low recall when demultiplexing data with high sequencing errors, with the recall for some samples falling below 80% (Fig. 5d), ~20% lower than TDFPS-Designer. This further suggests that Guppy struggles to handle sequencing errors effectively. More in-depth analysis reveals that Guppy classifies a large number of samples as “unclassified” under both types of sequencing data (Fig. 5e). This is because Guppy retains only the least ambiguous data, which ensures precision but causes a lot of data waste, whereas TDFPS-Designer effectively avoids this issue.

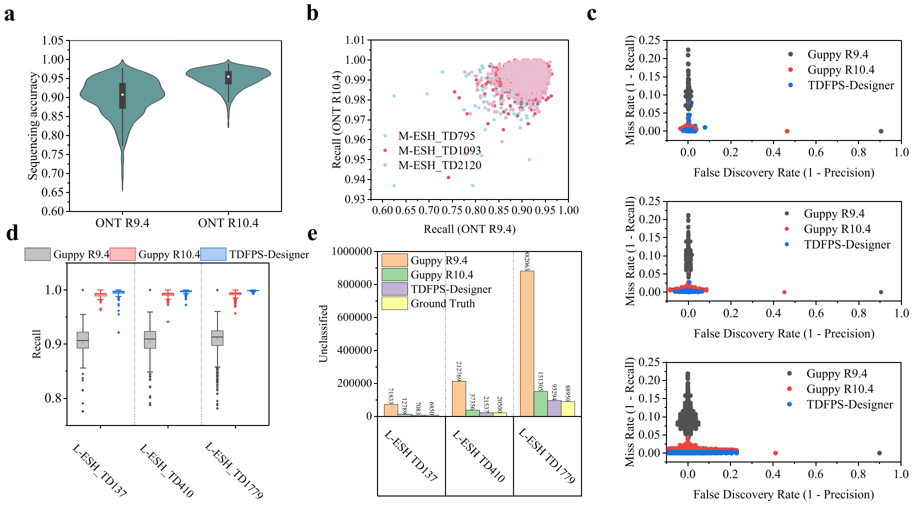


Fig. 5 Guppy’s and TDFPS-Designer’s demultiplexing analysis on different datasets with different sequencing error rates. **a** Sequencing accuracy distributions for two different error models. **b** Scatterplot of demultiplexing recall for Guppy on datasets with different sequencing error rates. “Guppy R9.4” (“Guppy R10.4”) indicates that multi-sample sequencing data generated by ONT R9.4 (R10.4) were demultiplexed using Guppy. **c** False discovery rate (1 - precision) vs. miss rate (1 - recall) scatterplot of Guppy and TDFPS-Designer on L-ESH_TD137 (top), L-ESH_TD410 (middle) and L-ESH_TD1779 (bottom). For better visualization, all overlapping points have been expanded horizontally. **d** Boxplot of Guppy’s and TDFPS-Designer’s demultiplexing recall on different datasets. **e** Bar charts of the unclassified values for Guppy and TDFPS-Designer on different datasets

Despite the continuous improvements in ONT sequencing accuracy, uncertainties still shroud the sequencing error rate, particularly in the context of nonmodel organisms and RNA samples [41]. In these scenarios, our demultiplexing approach emerges as a viable alternative solution.

Discussion

In nanopore sequencing, pooling multiple samples together for sequencing can save time and cost. However, separating raw sequencing data from multiple samples can be challenging. Barcodes are crucial for this purpose, while ONT provides barcode kits that support simultaneous sequencing of up to 96 samples. To enable simultaneous sequencing of more samples, we propose TDFPS-Designer, a new tool for designing barcodes using the TDFPS algorithm. The TDFPS algorithm improves the farthest point sampling algorithm. It uses the DTW distance as a measurement and a well-designed threshold to reduce the sampling space. Based on the TDFPS algorithm, TDFPS-Designer selects sequences that are sufficiently different from each other in the sequence space to construct barcode sets with different length. For the barcode kit, TDFPS-Designer has an efficient demultiplexing strategy, starting directly from the DTW distance matrix and completing the demultiplexing process, which ensures that the demultiplexing F1-score of all barcodes is above 95%. Additionally, TDFPS-Designer adopts a GPU acceleration mechanism to improve the efficiency of demultiplexing and barcode design.

Although Guppy is the current state-of-the-art tool for demultiplexing problems, experiments have shown that Guppy's demultiplexing performance is very susceptible to sequencing errors. In contrast, our method effectively overcomes this challenge, offering users a dependable demultiplexing solution for handling extensive sample demultiplexing issues. Our proposed barcode design strategy can design more barcodes while ensuring a stable demultiplexing effect, indicating that TDFPS-Designer has great development potential. To further enhance the performance of TDFPS-Designer, we plan to investigate more accurate barcode extraction strategies that can improve the accuracy of demultiplexing. This will be a focus of our future work.

Conclusions

In this study, we developed TDFPS-Designer, a new tool for designing barcodes using the TDFPS algorithm. The TDFPS algorithm enhances the farthest point sampling algorithm by employing the DTW distance as a measurement and implementing a well-designed threshold to minimize the sampling space. This method ensures that the sequences selected for barcode kits are sufficiently different from one another, enabling the construction of barcode kits with various lengths. Notably, the barcode kits designed by TDFPS-Designer are nearly 1.4 to 18.5 times larger than those provided by ONT, supporting the design of barcodes with arbitrary lengths. Experimental results demonstrate that the barcodes designed by TDFPS-Designer exhibit greater robustness compared to randomly generated barcodes. Moreover, the demultiplexing strategy employed by TDFPS-Designer is more effective in handling sequencing errors. Notably, under the condition of maintaining high demultiplexing accuracy, the recall rate of TDFPS-Designer is approximately 20% higher than that of Guppy. This suggests that the DTW algorithm in TDFPS-Designer is well-suited for handling the more common insertions and deletions in ONT, thereby ensuring a higher recall rate. This improvement ensures the feasibility and reliability of current multi-sample sequencing applications in non-model organisms and direct RNA sequencing.

Methods

TDFPS-Designer is developed using Python and C++. The primary function of this software is to design barcodes for ONT sequencing, facilitating the barcoding of a larger number of samples and enabling efficient demultiplexing. The use of TDFPS-Designer is described comprehensively in Additional File 1: S2 and https://github.com/junhaiqi/TDFPS Designer.git. Next, we provide details for each part of TDFPS-Designer.

Barcode design strategy: the maximum capacity of the barcode kit

Given a demultiplexing system S, dataset D, and an accuracy value p_{acc}, we define the barcode kit as BK, and the dataset D integrates BK for multisample sequencing as D_{BK}. $p_{D_{BK}}^{min}$ represents the minimum accuracy of the demultiplexing system S under D, where the minimum accuracy is defined in the "Evaluation metrics" section. For a dataset D, if the demultiplexing performance of S on D only depends on $|BK|$ (the size of BK), then there is a maximum capacity in theory:

$$C(D,p) = max(\{|BK| \; |BK, p_{D_{BK}}^{min} > p_{acc}\}). \tag{1}$$

TDFPS-Designer tries to find the BK with a demultiplexing capacity close to the maximum capacity. If a brute force scheme is adopted, we need to find all possible BK and calculate $p_{D_{BK}}^{min}$, which is obviously an NP-hard problem. It is presumed that there should be relatively large differences between the barcodes in the BK with the maximum capacity to facilitate demultiplexing. TDFPS-Designer uses the DTW distance to specify the barcode differences.

Barcode design strategy: selection of barcodes

Our barcode design strategy supports two input modes: the sequence length of the barcode kit and a given set of sequences of the same length. These input modes determine the unique sequence space from which we spatially pick sequences to serve as barcodes. Unfortunately, the sequence space can be very large. For example, there are over one million (4^{10}) choices within a barcode space of 10 bp barcode length and 109.9 billion (4^{20}) choices within a barcode space of 20 bp barcode length. To improve computational efficiency, we apply a simple initial selection scheme when the sequence space exceeds 1 million. In addition, our algorithm supports filtering out certain sequences when determining the sampling space to design barcodes that meet specific biological criteria, these biological criteria include balanced guanine-cytosine (GC) content, minimal homopolymer runs, and no self-complementarity of more than two bases to reduce internal hairpin propensity [24]. Figure 1a shows an illustration of this scheme. We define a hash function H on the nucleotide alphabet $\sum = \{A, T, C, G\}$ of DNA sequences, where $H(A) = 0$, $H(C) = 1$, $H(G) = 2$, and $H(T) = 3$. We extend this function to DNA sequences, as defined in Eq. (2):

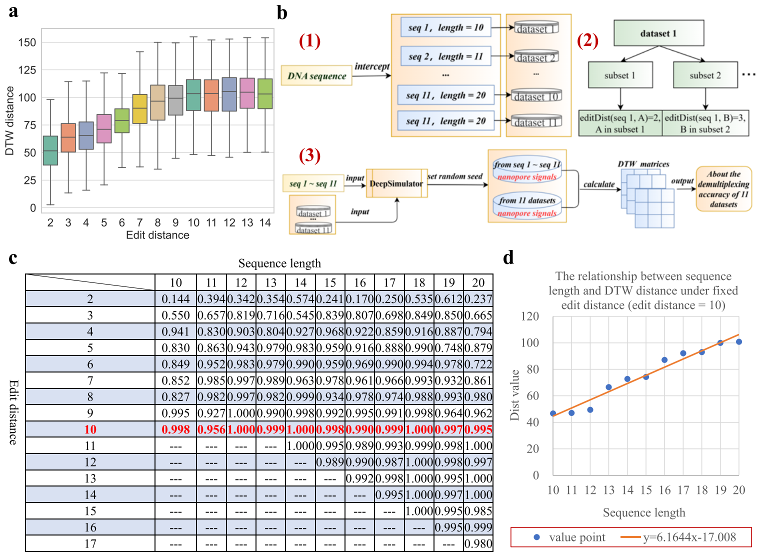
$$H(S) = H(s_1) \times 4^{k-1} + H(s_2) \times 4^{k-2} + ... + H(s_n), \tag{2}$$

where $S = s_1s_2...s_n$ represents a DNA sequence of length n.

Equation (2) reflects the relationship between sequences and their corresponding hash values. The greater the difference in hash values, the higher the probability that the two sequences have differences. We use this relationship to determine our initial selection strategy. We calculate and sort the hash values of all sequences and then use uniform random sampling to select one million items. We then select the sequences corresponding to these items to build the initial set of sequences. The final designed barcodes will all come from this initial set. Uniform random sampling selects samples across the entire range of sorted sequences, increasing the differences between the selected sequences. Uniform distribution in sampling reduces the probability of selecting similar (or adjacent) sequences, thereby enhancing the diversity of the sampled sequences.

To select the initial barcode set from the initially screened sequence set, we use a combination (called TDFPS algorithm) of the farthest point sampling algorithm and DTW algorithm and improve efficiency by incorporating a well-determined threshold r through experiments (in Fig. 6 below) . The goal is to ensure that the designed barcodes have enough differences to avoid sequencing errors affecting the demultiplexing results. We measure the difference between barcodes using the DTW distance between their corresponding signals. Specifically, the DTW distance between any two barcode signals in the final set should be greater than the threshold r.

Algorithm 1 outlines the selection of the initial barcode set. First, we convert the DNA sequence collection into a set of standard nanopore signals by the function *seq2sig*. We define the procedure $DTWSetVersion$ to calculate the minimum DTW distance between a signal and a set of signals. A new signal is identified as a barcode signal if and only if the DTW distance between this signal and the barcode signal set is large enough. Selecting the barcode directly based on the farthest point sampling algorithm would require running the DTW algorithm $\sim n^3$ times, where n is the size of the signal set. When the candidate barcode set is very large, this approach would still require considerable computational resources. To overcome this limitation, we reduce the size of the signal set based on the threshold r. Whenever a new barcode signal is selected, if the DTW distance between the signal in the signal set and this new barcode signal is less than



	10	11	12	13	14	15	16	17	18	19	20
2	0.144	0.394	0.342	0.354	0.574	0.241	0.170	0.250	0.535	0.612	0.237
3	0.550	0.657	0.819	0.716	0.545	0.839	0.807	0.698	0.849	0.850	0.665
4	0.941	0.830	0.903	0.804	0.927	0.968	0.922	0.859	0.916	0.887	0.794
5	0.830	0.863	0.943	0.979	0.983	0.959	0.916	0.888	0.990	0.748	0.879
6	0.849	0.952	0.983	0.979	0.990	0.959	0.969	0.990	0.994	0.978	0.722
7	0.852	0.985	0.997	0.989	0.963	0.978	0.961	0.966	0.993	0.932	0.861
8	0.827	0.982	0.997	0.982	0.999	0.934	0.978	0.974	0.988	0.993	0.980
9	0.995	0.927	1.000	0.990	0.998	0.992	0.995	0.991	0.998	0.964	0.962
10	**0.998**	**0.956**	**1.000**	**0.999**	**1.000**	**0.998**	**0.990**	**0.999**	**1.000**	**0.997**	**0.995**
11	---	---	---	---	1.000	0.995	0.989	0.993	0.999	0.998	1.000
12	---	---	---	---	---	0.989	0.990	0.987	1.000	0.998	0.997
13	---	---	---	---	---	---	0.992	0.998	1.000	0.995	1.000
14	---	---	---	---	---	---	---	0.995	1.000	0.997	1.000
15	---	---	---	---	---	---	---	---	1.000	0.995	0.985
16	---	---	---	---	---	---	---	---	---	0.995	0.999
17	---	---	---	---	---	---	---	---	---	---	0.980

Fig. 6 The process for determining the threshold *r*. **a** Distribution of DTW distances between nanopore signals corresponding to sequences. **b** The workflow for synthetic data generation. **c** Summary of demultiplexing accuracy at different sequence lengths and edit distances. **d** The linear relationship between sequence length and DTW distance

threshold *r*, it will be deleted, which could greatly reduce the size of the signal set and improve the screening efficiency. We also accelerate the calculation efficiency of the DTW matrix using CUDA and the diagonal parallel method, which improves the calculation efficiency of the DTW by $\sim$ 3 orders of magnitude [40].

Algorithm 1 Get the final barcodes from the initial sequence set

```
Input  : initial sequence set $initSet$, DTW distance threshold $t$
Output: final barcode set $barcodeSet$
$initNanoSigSet \leftarrow seq2sig(initSet)$ ;    // Conversion of DNA sequences into nanopore signals
$initNanoSig \leftarrow randomSelect(initNanoSigSet), B \leftarrow \{initNanoSig\}$ ;    // Randomly choose a signal
while $initNanoSigSet \neq \emptyset$ do
    $initNanoSigSet, selectedBarcode \leftarrow TDFPS(B, t, initNanoSigSet)$;
    $B \leftarrow B \cup \{selectedBarcode\}$;
$barcodeSet \leftarrow sig2seq(B)$ ;    // Based on the index to obtain barcode set.
Procedure $DTWSetVersion(sig, B)$:
    $sig2SetDist \leftarrow \infty$
    foreach $item \in B$ do
        if $sig2SetDist \geq DTW(item.sig)$ then
            $sig2SetDist \leftarrow DTW(item.sig)$
    return $sig2SetDist$
Procedure $TDFPS(B, t, initSigSet)$:
    $filteredSet \leftarrow \emptyset, distCutoff \leftarrow 0$
    foreach $sig \in initSigSet$ do
        $tempDist \leftarrow DTWSetVersion(sig, B)$    // The DTW distance between $sig$ and $B$.
        if $tempDist \leq t$ then
            $filteredSet \leftarrow filteredSet \cup \{sig\}$
        else
            if $tempDist \geq distCutoff$ then
                $selectedBarcode \leftarrow sig$    // Pick out the signal farthest from $B$.
                $distCutoff \leftarrow tempDist$
    $initSigSet \leftarrow initSigSet \setminus filteredSet$    // Filter out redundant signals.
    return $filteredSet, selectedBarcode$
```

We selected the final barcode kits from the initial barcode set (Fig. 1c). The initial barcodes exhibited high DTW dissimilarity, ensuring they could be easily distinguished. To further enhance the robustness of demultiplexing these barcodes, TDFPS-Designer ultimately screened the final barcodes by simulating a demultiplexing pipeline. Specifically, after the user specifies the sequencing platform (e.g., MinION R9.4 or MinION R10.4) and the multi-sample sequencing library information (adapter sequences and flanking sequences), all barcodes in the initial set are automatically used to construct a multi-sample sequencing library and generate a small batch of sequencing data. This sequencing data is then automatically demultiplexed by TDFPS-Designer. Subsequently, TDFPS-Designer analyzes the demultiplexing results, calculating the demultiplexing precision, recall, and F1-Score for each barcode. Barcodes with low precision (recall and F1-score) suggest potential conflicts with other barcodes in the kit, and TDFPS-Designer filters these out to obtain the final barcode kit.

Barcode design strategy: threshold determination

In theory, the demultiplexing accuracy depends on the difference between barcodes. Here, we want to determine a DTW distance through experimentation so that under this distance, a simple demultiplexing scheme can achieve sufficient precision. The determined distance threshold r is used as the termination condition of the TDFPS algorithm (Fig. 1a).

We generate template sequences of different lengths (ranging from 10 bp to 20 bp) for a given DNA sequence. By specifying an edit distance d, we generate 1000 sequences from these templates, where the edit distance between each generated sequence and its corresponding template sequence is d. As the DTW distance is correlated with the edit distance, larger editing distances between DNA sequences correspond to larger DTW distances between the corresponding nanopore signals (Fig. 6a). For each template sequence, we generate a dataset containing subsets of sequences with different edit distances from the template sequence (Fig. 6b: (1) and (2)). Using *DeepSimulator1.5* [42], we simulate nanopore signals from each template sequence and its corresponding dataset, calculate the DTW distance matrix between the template signal and signals in the dataset, and identify the demultiplexed result based on the row index of the smallest element in each column of the matrix. As shown in Fig. 6c, the demultiplexing accuracy exceeds 99% when the edit distance is 10 under different sequence lengths, indicating that the difference between barcodes is large enough. Moreover, we analyse the numerical distribution of the DTW distance for an edit distance of 10 under different sequence lengths and determine a linear function that determines the corresponding threshold (Fig. 6d).

Demultiplexing strategy

Figure 1b outlines our demultiplexing strategy. The first step involves detecting the barcode region in the nanopore signal. We design a heuristic strategy based on Oxford Nanopore's official multisample sequencing library construction scheme and the semiglobal DTW algorithm [43] to extract the barcode signal. This strategy involves detecting the region of the adapter signal to determine the position of the barcode signal and estimating the length of the barcode signal. Specifically, we assume that the sequence length of the barcode is n (excluding flanking sequences), and the estimated barcode signal length is $10n + c$, where c defaults to 70, based on the structural division of the nanopore signal (see Fig. 2a and b).

After extracting the barcode signals, we calculate the DTW distance matrix between these sequenced signals and the standard barcode signals, and the row index of the minimum value in each column of the distance matrix corresponds to the demultiplexed result. Specifically, upon extracting the minimum value from each row of the distance matrix, we employ the $5-\sigma$ method to detect anomalies. Any signals with a distance exceeding the threshold of $mean+5\times std$ are classified as anomalous data, potentially devoid of associated barcodes. Here, *mean* and *std* denote the mean and standard deviation of all distances, respectively.

Determination of final barcode kits

We used TDFPS-Designer to design final kits with barcodes of different lengths: 20 bp, 24 bp, and 30 bp, resulting in 137, 410, and 1779 barcodes, respectively. Specifically, we first designed 795, 1093, and 2120 barcodes of 20 bp, 24 bp, and 30 bp, respectively, based on the TDFPS algorithm. These barcodes ensure sufficient DTW distance differences, forming initial barcode kits. We used these barcode kits to generate three medium-sized datasets (M-ESH_TD795, M-ESH_TD1093, M-ESH_TD2120). We then demultiplexed these datasets. Figure 7a shows the distribution of demultiplexing recall. We can see that there is a positive correlation between the demultiplexing recall and the barcode length, indicating that the maximum capacity of the barcode kit is positively correlated with the barcode length. Additionally, we delved into the relationship between the number of barcodes and the minimum recall, which directly affects the estimation of the maximum capacity of the barcoded kit (as shown in Fig. 7b). It can be seen in Fig. 7b that once the number of barcodes exceeds a certain threshold, the minimum recall will drop significantly. This drop means that there will be "collisions" between certain barcodes, meaning that the demultiplexing system will have difficulty distinguishing certain barcodes accurately. To address this issue, TDFPS-Designer can simulate the generation of small batches of multi-sample sequencing data based on the initial barcode kit, automatically perform demultiplexing, and select the final barcodes from the initial barcode kit based on the demultiplexing results. These barcodes ensure > 95% precision, recall, and F1-Score during this process, forming final barcode kits. All parameters and corresponding output files are available at [44].

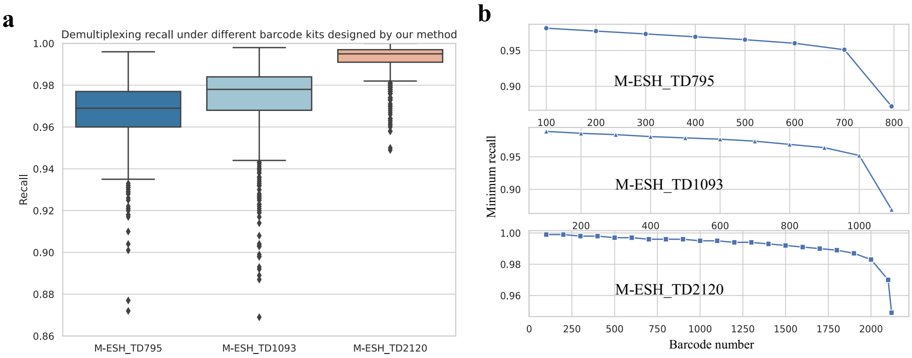


Fig. 7 Demultiplexing analysis of TDFPS-Designer on three medium-sized datasets (M-ESH_TD795, M-ESH_TD1093, and M-ESH_TD2120). **a** Boxplots of the demultiplexing accuracy of TDFPS-Designer on three medium-sized datasets. **b** Line graph between the lowest demultiplexing accuracy and the number of barcodes

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s13059-024-03423-3.

Additional file 1: S1. The relationship between the randomly selected sequence and the final set of barcodes. S2. Usage of TDFPS-Designer. S3. Evaluate simulated signals. S4. Detailed description of the dataset.

Additional file 2: Review history.

Acknowledgements
The authors thank the reviewers and editors for their valuable feedback.

Review history
The review history is available as Additional file 2.

Peer review information
Andrew Cosgrove was the primary editor of this article at Genome Biology and managed its editorial process and peer review in collaboration with the rest of the editorial team.

Authors' contributions
G. L., X. G., and R. H. conceived and managed the project. J. Q. implemented the algorithm. J. Q. and Z. L. collected all the datasets and performed all the analysis. R. H., G. L., X. G., and Y. Z. were involved in algorithm analysis and data analysis. All authors have read and approved the final manuscript.

Funding
This research was supported by the National Key Research and Development Program of China (2020YFA0712400 and 2021YFF0704300), the National Natural Science Foundation of China Projects Grant (62072280, 11931008, 61771009, 32241027), the Natural Science Foundation of Shandong Province ZR2023YQ057, the King Abdullah University of Science and Technology (KAUST) Office of Research Administration (ORA) under Award No FCC/1/1976-44-01, FCC/1/1976-45-01, REI/1/5234-01-01, REI/1/5414-01-01, URF/1/4352-01-01, and the open project of BGI-Shenzhen BGIRSZ20220005.

Data availability
All Python/C++ code of TDFPS-Designer is published under the permissive MIT open source license and is available on GitHub at https://github.com/junhaiqi/TDFPSDesigner.git. Additionally, the source code for TDFPS-Designer has been deposited at Zenodo [45]. All sequences used to generate simulated data are in [46–48], and the codes are in https://github.com/junhaiqi/MSNANOSIM and https://github.com/JustLeeee/ONT-sequencing-data-library-preparation-pipeline.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 14 November 2023 Accepted: 17 October 2024
Published online: 04 November 2024

References

1. Senol Cali D, Kim JS, Ghose S, Alkan C, Mutlu O. Nanopore sequencing technology and tools for genome assembly: computational analysis of the current state, bottlenecks and future directions. Brief Bioinforma. 2019;20(4):1542–59.
2. Choi JY, Lye ZN, Groen SC, Dai X, Rughani P, Zaaijer S, et al. Nanopore sequencing-based genome assembly and evolutionary genomics of circum-basmati rice. Genome Biol. 2020;21:1–27.
3. Shafin K, Pesout T, Lorig-Roach R, Haukness M, Olsen HE, Bosworth C, et al. Nanopore sequencing and the Shasta toolkit enable efficient de novo assembly of eleven human genomes. Nat Biotechnol. 2020;38(9):1044–53.
4. Moss EL, Maghini DG, Bhatt AS. Complete, closed bacterial genomes from microbiomes using nanopore sequencing. Nat Biotechnol. 2020;38(6):701–7.
5. Wang Y, Zhao Y, Bollas A, Wang Y, Au KF. Nanopore sequencing technology, bioinformatics and applications. Nat Biotechnol. 2021;39(11):1348–65.
6. Xie H, Li W, Hu Y, Yang C, Lu J, Guo Y, et al. De novo assembly of human genome at single-cell levels. Nucleic Acids Res. 2022;50(13):7479–92.
7. Fang Y, Chen G, Chen F, Hu E, Dong X, Li Z, et al. Accurate transcriptome assembly by Nanopore RNA sequencing reveals novel functional transcripts in hepatocellular carcinoma. Cancer Sci. 2021;112(9):3555–68.
8. Sahlin K, Medvedev P. Error correction enables use of Oxford Nanopore technology for reference-free transcriptome analysis. Nat Commun. 2021;12(1):2.

9. de la Rubia I, Srivastava A, Xue W, Indi JA, Carbonell-Sala S, Lagarde J, et al. RATTLE: reference-free reconstruction and quantification of transcriptomes from Nanopore sequencing. Genome Biol. 2022;23(1):153.
10. Liu Y, Rosikiewicz W, Pan Z, Jillette N, Wang P, Taghbalout A, et al. DNA methylation-calling tools for Oxford Nanopore sequencing: a survey and human epigenome-wide evaluation. Genome Biol. 2021;22(1):1–33.
11. Tourancheau A, Mead EA, Zhang XS, Fang G. Discovering multiple types of DNA methylation from bacteria and microbiome using nanopore sequencing. Nat Methods. 2021;18(5):491–8.
12. Sakamoto Y, Zaha S, Nagasawa S, Miyake S, Kojima Y, Suzuki A, et al. Long-read whole-genome methylation patterning using enzymatic base conversion and nanopore sequencing. Nucleic Acids Res. 2021;49(14):e81–e81.
13. Cumbo C, Minervini CF, Orsini P, Anelli L, Zagaria A, Minervini A, et al. Nanopore targeted sequencing for rapid gene mutations detection in acute myeloid leukemia. Genes. 2019;10(12):1026.
14. Goenka SD, Gorzynski JE, Shafin K, Fisk DG, Pesout T, Jensen TD, et al. Accelerated identification of disease-causing variants with ultra-rapid nanopore genome sequencing. Nat Biotechnol. 2022;40(7):1035–41.
15. Capraru ID, Romanescu M, Anghel FM, Oancea C, Marian C, Sirbu IO, et al. Identification of Genomic Variants of SARS-CoV-2 Using Nanopore Sequencing. Medicina. 2022;58(12):1841.
16. Church GM, Kieffer-Higgins S. Multiplex DNA sequencing. Science. 1988;240(4849):185–8.
17. Wick RR, Judd LM, Holt KE. Deepbinner: Demultiplexing barcoded Oxford Nanopore reads with deep convolutional neural networks. PLoS Comput Biol. 2018;14(11):e1006583.
18. Smith MA, Ersavas T, Ferguson JM, Liu H, Lucas MC, Begik O, et al. Molecular barcoding of native RNAs using nanopore sequencing and deep learning. Genome Res. 2020;30(9):1345–53.
19. Whitford W, Hawkins V, Moodley K, Grant MJ, Lehnert K, Snell RG, et al. Optimised multiplex amplicon sequencing for mutation identification using the MinION nanopore sequencer. bioRxiv. 2021;2021–09.
20. Peterson WW, Peterson W, Weldon EJ, Weldon EJ. Error-correcting codes, vol 2. Cambridge: MIT Press google schola; 1972. p. 208–213.
21. MacWilliams FJ, Sloane NJA. The theory of error-correcting codes, vol 2. Elsevier Science Publishers BV google schola; 1977. p. 9–47.
22. Hamady M, Walker JJ, Harris JK, Gold NJ, Knight R. Error-correcting barcoded primers for pyrosequencing hundreds of samples in multiplex. Nat Methods. 2008;5(3):235–7.
23. Zorita E, Cusco P, Filion GJ. Starcode: sequence clustering based on all-pairs search. Bioinformatics. 2015;31(12):1913–9.
24. Hawkins JA, Jones SK Jr, Finkelstein IJ, Press WH. Indel-correcting DNA barcodes for high-throughput sequencing. Proc Natl Acad Sci. 2018;115(27):E6217–26.
25. Doroschak K, Zhang K, Queen M, Mandyam A, Strauss K, Ceze L, et al. Rapid and robust assembly and decoding of molecular tags with DNA-based nanopore signatures. Nat Commun. 2020;11(1):5454.
26. Marzal A, Vidal E. Computation of normalized edit distance and applications. IEEE Trans Pattern Anal Mach Intell. 1993;15(9):926–32.
27. Kelley DR, Schatz MC, Salzberg SL. Quake: quality-aware detection and correction of sequencing errors. Genome Biol. 2010;11:1–13.
28. Galanti L, Shasha D, Gunsalus KC. Pheniqs 2.0: accurate, high-performance Bayesian decoding and confidence estimation for combinatorial barcode indexing. BMC Bioinforma. 2021;22:1–16.
29. Lu G, Zhang S, Fang X. An improved string composition method for sequence comparison. BMC Bioinforma. 2008;9(6):1–8.
30. Reinert G, Chew D, Sun F, Waterman MS. Alignment-free sequence comparison (I): statistics and power. J Comput Biol. 2009;16(12):1615–34.
31. Aita T, Husimi Y, Nishigaki K. A mathematical consideration of the word-composition vector method in comparison of biological sequences. BioSystems. 2011;106(2–3):67–75.
32. Dai Q, Liu X, Yao Y, Zhao F. Numerical characteristics of word frequencies and their application to dissimilarity measure for sequence comparison. J Theor Biol. 2011;276(1):174–80.
33. Papetti DM, Spolaor S, Nazari I, Tirelli A, Leonardi T, Caprioli C, et al. Barcode demultiplexing of nanopore sequencing raw signals by unsupervised machine learning. Front Bioinforma. 2023;3:1067113.
34. Guan X, Li Z, Zhou Y, Shao W, Zhang D. Active learning for efficient analysis of high-throughput nanopore data. Bioinformatics. 2023;39(1):btac764.
35. Loose M, Malla S, Stout M. Real-time selective sequencing using nanopore technology. Nat Methods. 2016;13(9):751–4.
36. Han R, Li Y, Gao X, Wang S. An accurate and rapid continuous wavelet dynamic time warping algorithm for end-to-end mapping in ultra-long nanopore sequencing. Bioinformatics. 2018;34(17):i722–31.
37. Han R, Wang S, Gao X. Novel algorithms for efficient subsequence searching and mapping in nanopore raw signals towards targeted sequencing. Bioinformatics. 2020;36(5):1333–43.
38. Gamaarachchi H, Ferguson JM, Samarakoon H, Liyanage K, Deveson IW. Simulation of nanopore sequencing signal data with tunable parameters. Genome Res. 2024;34(5):778–83.
39. Wick RR. Badread: simulation of error-prone long reads. J Open Source Softw. 2019;4(36):1316.
40. Han R, Qi J, Xue Y, Sun X, Zhang F, Gao X, et al. HycDemux: a hybrid unsupervised approach for accurate barcoded sample demultiplexing in nanopore sequencing. Genome Biol. 2023;24(1):1–29.
41. Liu-Wei W, van der Toorn W, Bohn P, Hölzer M, Smyth RP, von Kleist M. Sequencing accuracy and systematic errors of nanopore direct RNA sequencing. BMC Genomics. 2024;25(1):528.
42. Li Y, Wang S, Bi C, Qiu Z, Li M, Gao X. DeepSimulator1. 5: a more powerful, quicker and lighter simulator for Nanopore sequencing. Bioinformatics. 2020;36(8):2578–80.
43. Boža V, Brejová B, Vinař T. Improving Nanopore Reads Raw Signal Alignment. arXiv preprint arXiv:1705.01620. 2017;2017-05.
44. Qi J. TDFPS-Designer: an efficient toolkit for barcode design and selection in nanopore sequencing. 2024. Zenodo. https://doi.org/10.5281/zenodo.13927379.
45. Qi J. TDFPS-Designer: an efficient toolkit for barcode design and selection in nanopore sequencing. 2024. Zenodo. https://doi.org/10.5281/zenodo.8260659.

46. Li Z. TDFPS-Designer: an efficient toolkit for barcode design and selection in nanopore sequencing. 2024. Zenodo. https://doi.org/10.5281/zenodo.13208175.
47. Li Z. TDFPS-Designer: an efficient toolkit for barcode design and selection in nanopore sequencing. 2024. Zenodo. https://doi.org/10.5281/zenodo.13203290.
48. Li Z. TDFPS-Designer: an efficient toolkit for barcode design and selection in nanopore sequencing. 2024. Zenodo. https://doi.org/10.5281/zenodo.13923770.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.