To address the insertion and deletion errors introduced by nanopore sequencing technology in DNA data storage, as well as the high complexity of sequence assembly techniques in data recovery process, we propose a pragmatic soft-decision data readout method that enables assembly-free sequence reconstruction, indel error correction, and ultra-low coverage data readout.

Here we provide the code of the soft-decision data readout pipeline used for encoded large DNA. To enhance understanding and better monitor the running of our code, the process of data readout is assigned into four programs with chained input and output, each corresponding to the core components of the proposed method: (1) watermark alignment; (2) soft-decision forward-backward algorithm; (3) soft information merging from multiple reads; and (4) soft-decision decoding. These input and output files are provided along with the programs. The software is implemented in C/C++, and our executable calls are integrated into shell scripts, allowing for easy and quick deployment on different versions of Linux systems.

In our work, to demonstrate the feasibility of the proposed method, the third-generation nanopore sequencer was used for the attempt to retrieve the files from a set of large DNA fragments (two plasmids of ~51k bp, a yeast artificial chromosome of 254,886 bp). Here we provide the sequencing data, as well as data readout-related files and programs, for one of the plasmids (length: 51,339 bp).

**NOTE:** For detailed information on coding procedures and sequencing data for yeast artificial chromosomes, see the literature: Chen W, Han M, Zhou J, Ge Q, Wang P, Zhang X, et al. An artificial chromosome for data storage. Natl Sci Rev 2021;8:nwab086. <https://doi.org/10.1093/nsr/nwab028>. In addition, the program used for data assembly can be obtained from the following URL. In fact, any assembly software meeting the requirements can be used for sequencing read assembly. Other computer codes used in this work are available from the authors upon request.

Minimap2: https://github.com/lh3/minimap2

Miniasm: https://github.com/lh3/miniasm

RACON: https://github.com/isovic/racon

**Table 1** lists the openly available plasmid sequence, watermark sequence, and sequencing data used in our work.

**Table 2** lists the programs required for the complete readout process.

**Table 1. Plasmid sequence, watermark sequence, and sequencing data used in our study.**

|  |  |  |
| --- | --- | --- |
| **Files** | **Storage location** | Description |
| genomefile\_A.txt | current catalog | Encoded payload sequence, length: 40,500 nucleotides |
| TJ0083169-1-plasmid-A.txt | current catalog | Designed plasmid sequence of 51,339 bp, used as reference, containing payload and plasmid vector |
| SequenceLengthALL\_FILE001R025 | current catalog | Watermark sequence |
| plasmid-A-head.txt | current catalog | Plasmid vector sequences |
| plasmid-A-tail.txt |
| Guppy\_barcode02.zip | nanopore\_duplex\_fastq | Standard sequencing reads generated using Guppy single-strand basecalling software |
| dorado\_barcode02\_duplex.fastq | nanopore\_duplex\_fastq | Duplex sequencing reads generated using Dorado duplex basecalling software |

**Table 2 Programs used for data readout from nanopore sequencing reads and input/output files.**

|  |  |  |  |
| --- | --- | --- | --- |
| Program | Input | Output | Description |
| step1-run\_watermark\_alignment.sh | dorado\_barcode02\_duplex.fastq &  SequenceLengthALL\_FILE001R025 &  plasmid-A-head.txt &  plasmid-A-tail.txt | watermark\_alignment\_result\_output.txt | Step 1: Run the “watermarkAlignment” program to align sequencing reads with the aid of watermark. |
| step2-run\_soft\_decision\_FBA.sh | watermark\_alignment\_result\_output.txt &  TJ0083169-1-plasmid-A.txt &  SequenceLengthALL\_FILE001R025 &  Watermark length &  Ins &  Sub &  Del &  plasmid-A-head.txt &  plasmid-A-tail.txt | indelCorrect\_output\_Soft\_decision.txt | Step2: Run “random\_select\_waterAlign\_result” to randomly select some sequencing reads from the watermark alignment results;  Run the program “fb\_soft\_indel\_corrrect” to identify indel errors of sequencing reads and convert them into probability information that can be decoded. |
| step3-LLRCombine.sh | indelCorrect\_output\_Soft\_decision.txt &  genomefile\_A.txt &  SequenceLengthALL\_FILE001R025 | soft\_infor\_consensus.txt | Step 3: Run the “LLRmerging” program to obtain consensus soft information from multiple reads. |
| step4-decoding.sh | soft\_infor\_consensus.txt | plasmid\_A\_S\_correctedBitStream.txt &  Dreams.txt | Step 4: Run the “LDPC\_r1\_4\_decoder\_soft” program to complete soft-decision decoding. |

**Example of usage for our programs:**

* **Run “step1-run\_watermark\_alignment.sh”:**

# Run the shell script like this:

./**step1-run\_watermark\_alignment.sh**

# The description of input and output files:

**Input files:**

* dorado\_barcode02\_duplex.fastq: Duplex sequencing reads generated by Dorado software, used as raw data for data recovery.
* SequenceLengthALL\_FILE001R025: Watermark sequence embedded in the designed encoded large DNA fragment, used to assist in locating raw reads.
* plasmid-A-head.txt & plasmid-A-tail.txt: Vector parts of the plasmid sequence, used to distinguish the payload region from the vector region in sequencing reads.

**Output files:**

* watermark\_alignment\_result\_output.txt: The result of watermark alignment. Run “step1-xx.sh”, you will obtain the file named “watermark\_alignment\_result\_output.txt” containing the read alignment results, which is saved in the folder “watermark\_alignment/result/genome-A”. Each set of three lines contains the start position, length, and sequence of a read.
* **Run “step2-run\_soft\_decision\_FBA.sh”:**

# Run the shell script like this:

./**step2-run\_soft\_decision\_FBA.sh**

# The description of input and output files:

This script includes two functions: (1) randomly select a certain number of reads from the watermark alignment results for error correction; (2) identify indel errors in reads using the soft-decision forward-backward algorithm and output soft information for each encoded symbol.

**Function 1 input files:**

* watermark\_alignment\_result\_output.txt: The result generated after running “step 1”.
* readNum=30: The number of reads you want to select; here, “readNum=30” is used as an example.

**Function 1 output files:**

* downSample\_30.txt: File storing the randomly selected reads.

**Function 2 input files:**

* TJ0083169-1-plasmid-A.txt: Reference sequence of the plasmid, used to compare the residual substitution error rate after error correction.
* SequenceLengthALL\_FILE001R025: Watermark sequence embedded in the designed encoded large DNA fragment, used to assist in identifying indel errors in raw reads.
* Watermark length: Length of the watermark sequence.
* Ins & Sub & Del &: Parameters for insertion, substitution, and deletion error rates used in the Hidden Markov Model. These parameters are derived from observed sequencing read error characteristics.
* plasmid-A-head.txt & plasmid-A-tail.txt: Vector parts of the plasmid sequence, used to distinguish the payload region from the vector region in sequencing reads.

**Function 2 output files:**

* indelCorrect\_output\_Soft\_decision.txt: The result after identifying indel errors. Run “step2-xx.sh”, you will obtain the file “indelCorrect\_output\_Soft\_decision.txt” containing the soft information for each read after indel error correction, which is saved in the folder “soft\_decision\_FBA/result/genome-A”. Each set of three lines contains the start position, length, and soft information of each encoded symbol.
* **Run “step3-LLRCombine.sh”:**

# Run the shell script like this:

./**step3-LLRCombine.sh**

# The description of input and output files:

**Input files:**

* indelCorrect\_output\_Soft\_decision.txt: Soft information generated by program “step 2”.
* SequenceLengthALL\_FILE001R025: Watermark sequence embedded in the designed encoded large DNA fragment, used to XOR with sequencing data to remove the watermark.
* genomefile\_A.txt: Encoded payload sequence, used as a reference to verify the error rate of the results from program “step 3”.

**Output files:**

* soft\_infor\_consensus.txt: The result of indel correction. Run “step3-xx.sh”, you will obtain the consensus soft information file “soft\_infor\_consensus.txt”, which is saved in the folder “multiple\_read\_probability\_merging/result/genome-A”. The result contains two columns of probability information: the probability of “1” in the first column and the probability of “0” in the second column. This probability information will be used for soft-decision decoding.
* **Run “step4-decoding.sh”:**

# Run the shell script like this:

./**step4-decoding.sh**

# The description of input and output files:

**Input files:**

* soft\_infor\_consensus.txt: Probability information of bits “1” and “0” generated by program “step 3”.

**Output files:**

* plasmid\_A\_S\_correctedBitStream.txt: Decoded bit stream. Run “step4-xx.sh”, you will obtain the bitstream file “plasmid\_A\_S\_correctedBitStream.txt” containing the decoded results, which is saved in the folder “decoding\_process/result/genome-A”.
* Dreams.txt: Digital file recovered from the decoding result. In this example, the stored digital file is the poetry “Dreams”.