Mansoura University
Faculty of Computer and Information Sciences

Nanopore Sequencing Algorithm

DeepSimulator1.5

By/ Ayman Ahmed Bakshesh

Supervised by/ Prof.Sara El-Sayed El-Matwally

paper link: https://academic.oup.com/bioinformatics/article/36/8/2578/5698265

Subjects

1 - Workflow and implementation

2 - Abstract and Introduction summary

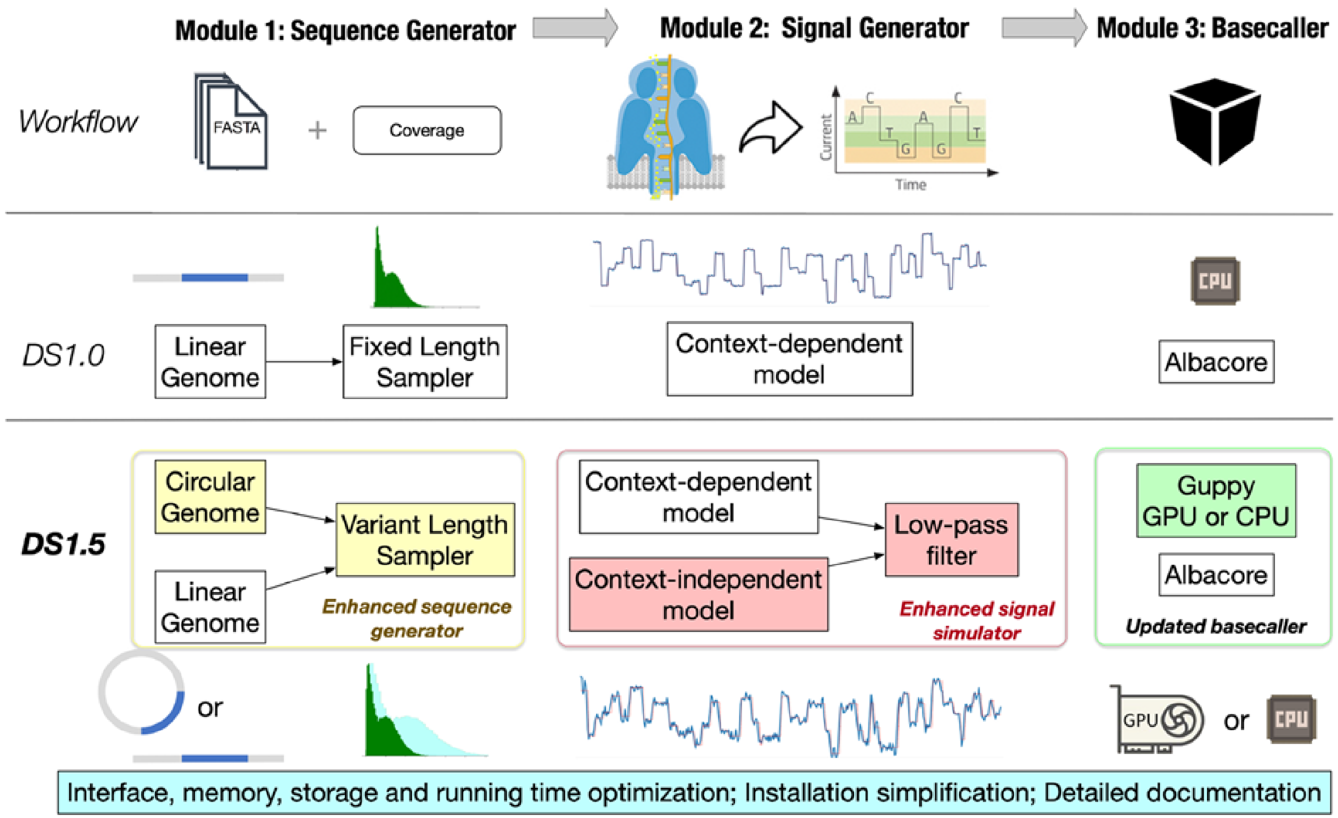
3 - Related works summary

4 - methodology

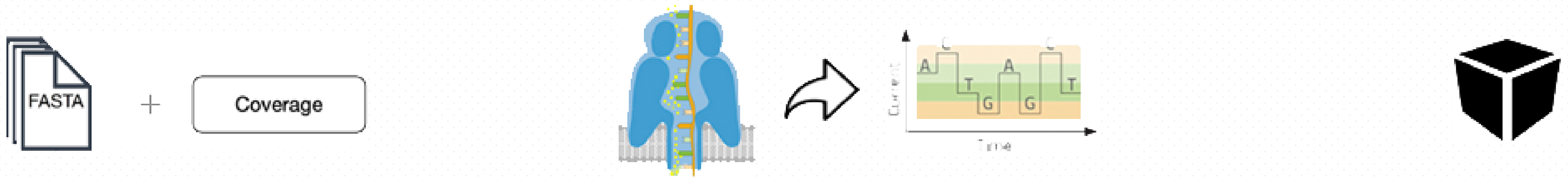
5 - result

DeepSimulator

1 – Workflow and implementation



Workflow

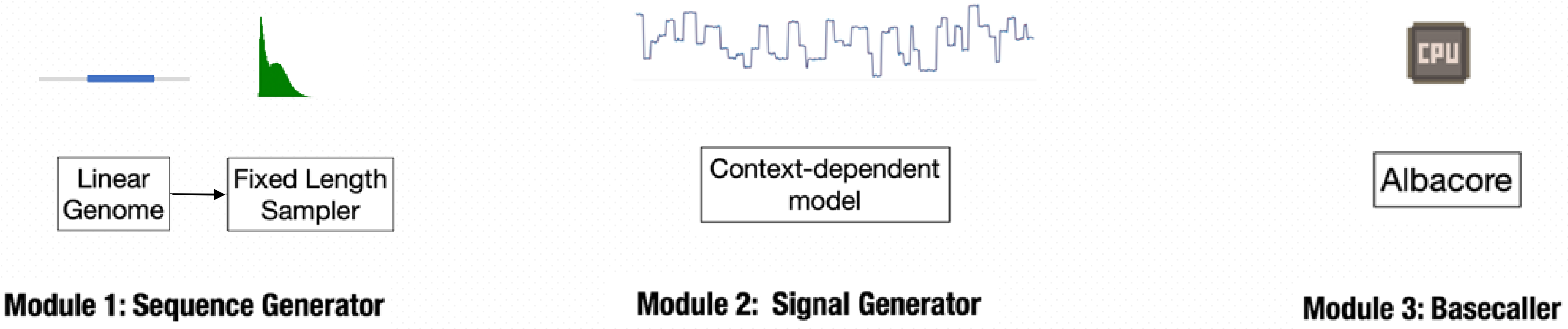


Module 1: Sequence Generator

Module 2: Signal Generator

Module 3: Basecaller

DS1.0



Linear Genome → Fixed Length Sampler

Module 1: Sequence Generator

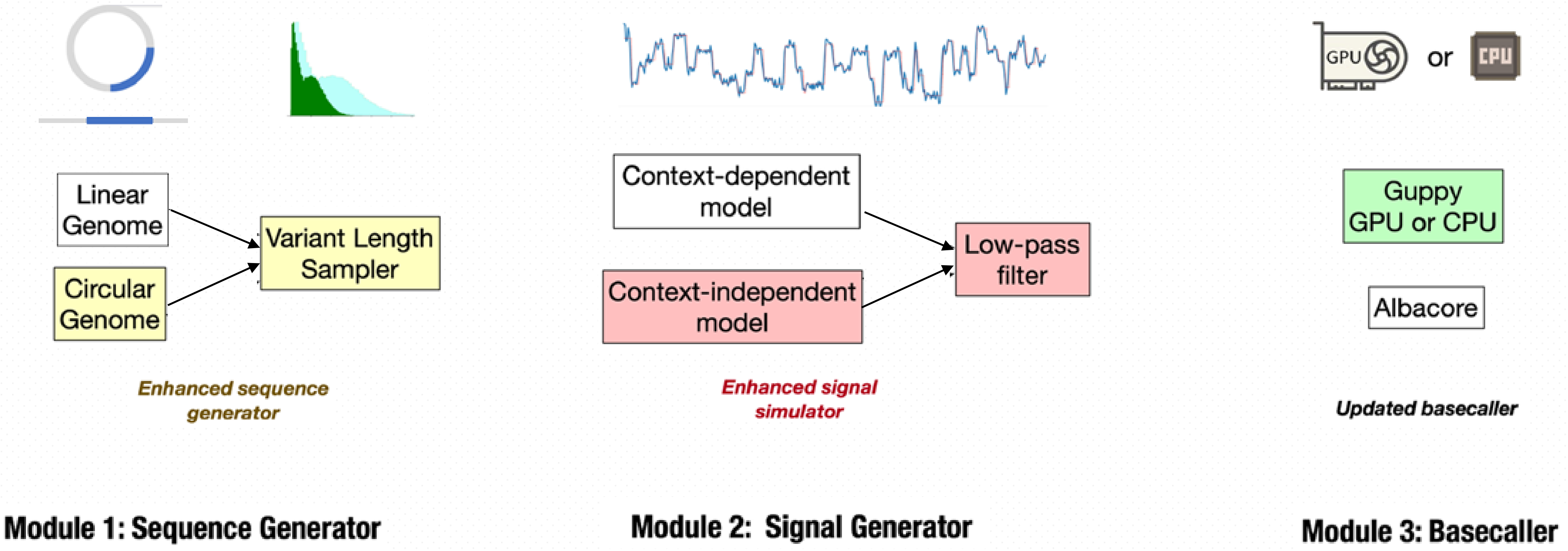
Context-dependent model

Module 2: Signal Generator

Albacore

Module 3: Basecaller

DS1.5



Linear Genome

Circular Genome

Variant Length Sampler

Enhanced sequence generator

Context-dependent model

Context-independent model

Low-pass filter

Enhanced signal simulator

Guppy GPU or CPU

Albacore

Updated basecaller

Module 1: Sequence Generator

Module 2: Signal Generator

Module 3: Basecaller

1 . 1 - Sequence generator

DS was designed to simulate the entire Nanopore sequencing procedure, including sequence generator, raw signal generator and basecaller. Given the target genome sequence, the sequence generator samples sequences from the genome, which correspond to the DNA segments that pass through the molecular pore in the real experiments. Although this module is conceptually simple, we have included the following updates into DS1.5 to meet the needs of different users. Previously, by default, this module can only sample the linear genome. Now, we equipped it with the power to sample the circular genome or generate the reads without sampling. Furthermore, based on the feedback of the users .

1 . 2 - Signal generator

The sampled sequences will go through the signal generator to output the simulated signals, whose behavior mimics that of a Nanopore sequencing device. In the signal generator, we use a deep learning-based pore model to produce the expected signals at each position of the input sequences. Then, each signal will be repeated several times based on the pattern in the real signals to produce the simulated signals .

1 . 3 – Basecaller

After obtaining the signals produced by the signal generator, the next step is to translate the signals into the final reads, which correspond to the final sequence outputs in the real experiment. Although the users can feed a customized basecaller to DS, based on our experience, the users tend to use the default basecaller. Previously, the default basecaller of DS1.0 is Albacore. In London Calling 2019 (LC19), the Nanopore Tech has officially released a more powerful basecaller, Guppy. To cope with this evolution, we added both the GPU and CPU versions of Guppy into DS1.5 and made the GPU one the default basecaller .

2 - Abstract and Introduction summary

- One of the most important revolutions in the field of biology was caused by the development of next-generation sequencing (NGS) technologies. Using massively parallel processing of samples, NGS dramatically reduces sequencing time and costs, enabling the sequencing of entire genomes. Currently, genome sequencing and analysis have become a crucial component in biology .

- The problem is the exponential increase of reported genomes on GenBank " a 6-fold increase in only 4 years "

- Thus,You must choose a more effective tool and method of obtaining to make the generated signals more similar to the real ones, we added a low-pass filter to post-process the pore model signals.

- To solve this problem we update all three modules from DS1.0. to DS1.5

- As for the sequence generator, we updated the sample read length distribution to reflect the newest real reads' features

- added a low-pass filter to post-process the pore model signals.

- added the support for the newest official basecaller, Guppy, which can support both GPU and CPU.

3 - Related works summary

- **simuG**
  - SimuGis a lightweight tool for simulating the full-spectrum of genomic variants (single nucleotide polymorphisms, Insertions/Deletions, copy number variants, inversions and translocations) .
  - Is a command-line tool written in Perl and supports all mainstream operating systems.
- **DeepSimulator**
  - DeepSimulator a tool for simulate the electrical current signals by a context-dependent deep learning model, followed by a base-calling procedure to yield simulated reads.
  - benefit the development of tools in de novo assembly and in low coverage SNP detection.
- **SiLiCO**
  - SiLiCO the first open source package for in silico simulation of long read sequencing results on both major long read sequencing platforms

4 - methodology

- **WaveNano**
  - bi-directional WaveNet model with residual blocks and skip connections is able to capture the extremely long dependency in the raw signal.

- **Nanopore sequencing technology and tools for genome assembly**
  - ONT’s basecalling tools, Metrichor, Nanonet and Scrappie, are the best choices for the basecalling step in terms of both accuracy and performance. Among these tools, Scrappie is the newest, fastest and most accurate basecaller. Thus, we recommend using Scrappie for the basecalling step

- **continuous wavelet dynamic time warping algorithm (DTW)**
  - algorithm starts from low-resolution wavelet transforms of the two sequences, such that the transformed sequences are short and have similar sampling rates
  - Then the peaks and nadirs of the transformed sequences are extracted to form feature sequences with similar lengths, which can be easily mapped by the original DTW .

5 - result

- In this work, we reported a new version of the previously published work on simulating the Nanopore sequencing, DeepSimulator1.5

- In this updated version, we have updated all the three modules of DeepSimulator significantly with several crucial overall optimizations, resulting in a more powerful, quicker and lighter simulator.

DeepSimulator

Thank You