Mansoura University
Faculty of Computer and Information Sciences

DeepSimulator

Nanopore Sequencing Algorithm

DeepSimulator1.5

By / Ayman Ahmed Bakshesh

Subjects

1 - Workflow and implementation

Sequence generator . Signal generator . Basecaller

2 - Abstract and Introduction summary

3 - Related works summary

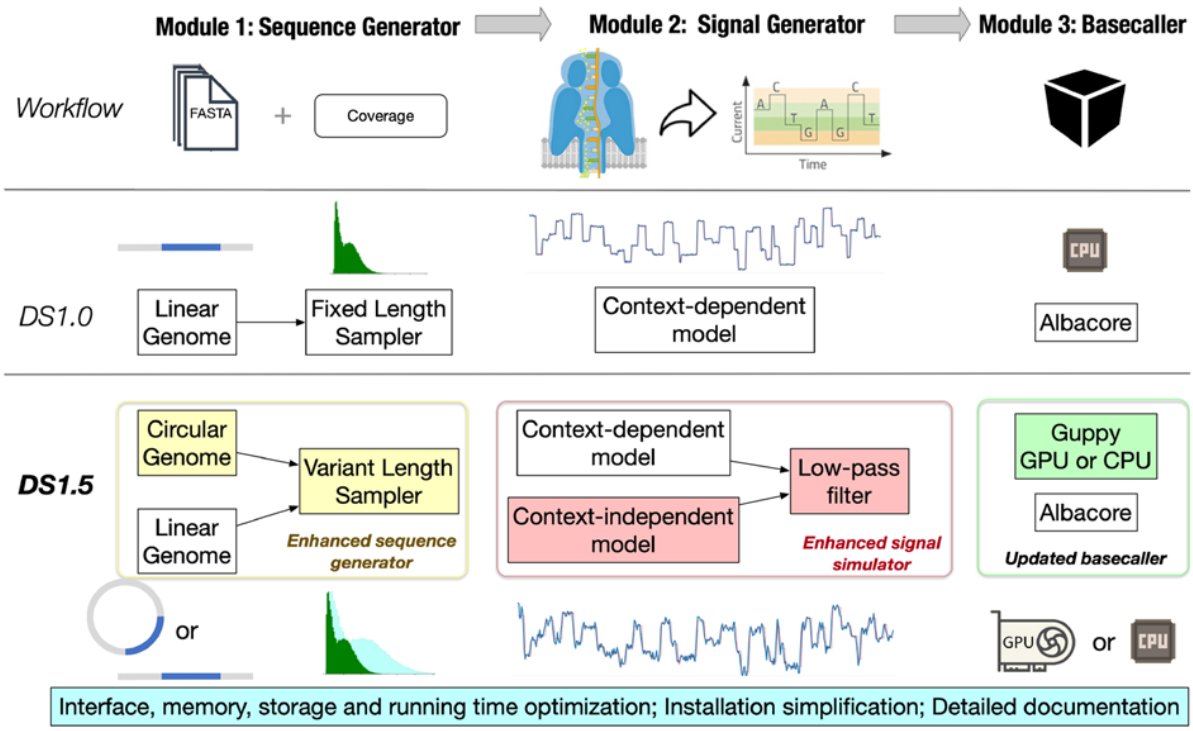
simuG . DeepSimulator . SiLiCO

4 - methodology

WaveNano . Nanopore sequencing technology . DTW

5 - result

1 - Workflow and implementation



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 (e.g., from 30,000 sequenced prokaryotic genomes in 2014 to 183,000 in 2018) ," 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 one more pore model, the context-independent pore model, which is much faster than the previous context-dependent one.

- 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.

- multiple optimizations, related to multiprocessing control, memory and storage management, have been implemented to make DS1.5 a much more amenable and lighter simulator than DS1.0.

- contents of paper will be Workflow , implementation , Performances , Conclusions and discussion of this update.

- Although the first version of DeepSimulator (DS1.0) has been recognized and used by a number of users " https://github.com/lykaust15/DeepSimulator " .

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.
  - Takes the user-supplied reference genome (in FASTA format) as the working template to introduce non-overlapping genomic variants of all major types (i.e. SNPs, INDELs, CNVs, inversions and translocations).
  - Code in Perl along with user manual and testing data is available at https://github.com/yjx1217/simuG.
  - This software is free for use under the MIT license.

- **DeepSimulator**
  - DeepSimulator, is mimic the entire pipeline of Nanopore sequencing. Starting from a given reference genome or assembled contigs, we simulate the electrical current signals by a context-dependent deep learning model, followed by a base-calling procedure to yield simulated reads.
  - The thorough experiments performed across four species show that the signals generated by our context-dependent model are more similar to the experimentally obtained signals than the ones generated by the official context-independent pore model.
  - so that they can obtain the reads with different accuracies ranging from 83 to 97%.
  - Two case studies demonstrate the application of DeepSimulator to 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.
  - SiLiCO simulates both PacBio and, for the first time, Oxford Nanopore read sequencing results by sto-chastically generating genomic coordinates and extracting corresponding nucleotide sequences from a reference assembly.
  - SiLiCO also is easily scaled up to a Monte-Carlo simulation, affording the end user the ability to construct empirical distributions of various genomic features.
  - SiLiCO is an open source package written in Python. It is freely available at https://www.github.com/ethanagbaker/SiLiCO under the GNU GPL 3.0 license.

4 - methodology and result

• Nanopore sequencing poses a number of computational challenges, for which various methods and algorithms have been developed

- **WaveNano**
  - show that the indel issue can be significantly reduced via accurate labeling of nucleotide and move labels directly from the raw signal, which can then be efficiently learned by a bi-directionalWaveNet model simultaneously through feature sharing
  - bi-directional WaveNet model with residual blocks and skip connections is able to capture the extremely long dependency in the raw signal.
  - Taking the predicted move as the segmentation guidance, we employ the Viterbi decoding to obtain the final base-calling results from the smoothed nucleotide probability matrix.

- **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 (see analysis in section 'Basecalling tools').
  - For the polishing step, we recommend using Racon, as it is much faster than Nanopolish. Racon also produces highly-accurate assemblies (see analysis in section 'Read mapping and polishing tools').
  - For the read-to-read overlap finding step, Minimap is faster than GraphMap, and it requires low memory. Also, it has similar accuracy to GraphMap. Thus, we recommend Minimap for the read-to-read overlap finding step (see analysis in section 'Read-to-read overlap finding tools').

- **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.
  - Our algorithm then recursively projects the warping path from a lower-resolution level to a higher-resolution one by building a context-dependent boundary and enabling a constrained search for the warping path in the latter.

……..

- **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.
  - This major update can remarkably broaden its applications in large-scale sequencing simulations as well as studies focusing on the Nanopore signals.