Realtime analysis and visualisation of MinION sequencing data with npReader

Minh Duc Cao *, Devika Ganesamoorthy, Matthew A. Cooper and Lachlan J. M. Coin *

Institute for Molecular Bioscience, The University of Queensland, Brisbane, Australia

ABSTRACT

Motivation: The recently released Oxford Nanopore MinION sequencing platform presents many innovative features opening up potential for a range of applications not previously possible. Among these features, the ability to sequence in real-time provides a unique opportunity for many time-critical applications. While many software packages have been developed to analyse its data, there is still a lack of toolkits that support the streaming and real-time analysis of MinION sequencing data.

Results: We developed npReader, an open-source software package to facilitate real-time analysis of MinION sequencing data. npReader can simultaneously extract sequence reads and stream them to downstream analysis pipelines while the samples are being sequenced on the MinION device. It provides a command line interface for easy integration into a bioinformatics work flow, as well as a graphical user interface which concurrently displays the statistics of the run. It also provides an application programming interface for development of streaming algorithms in order to fully utilize the extent of nanopore sequencing potential.

Availability and implementation: npReader is written in Java and is freely available at `https://github.com/mdcao/npReader`.

Contact: Minh Duc Cao (m.cao1@uq.edu.au) and Lachlan J. M. Coin (l.coin@imb.uq.edu.au).

1 INTRODUCTION

The recent introduction of the Oxford Nanopore MinION sequencing platform has added a new dimension into genomics sequencing. The platform can read ultra long DNA segments due to the ability to directly sense DNA fragments with biological nanopores without the need of amplification. Its portability also proves useful to many studies under critical conditions such as disease outbreaks and field work. A number of labs around the world have been evaluating the platform through the MinION Access Programme (MAP), and have also used the platform to develop numerous novel applications.

One of the most innovative features of the Oxford Nanopore MinION platform is that it can output sequence reads in real-time, *i.e.*, the electrical current data and the base-called sequence of a DNA fragment can be obtained as soon as it passed through the nanopore. This is attractive for many time critical applications

such as rapid infection diagnosis. While several excellent software packages (*e.g.*, Loman and Quinlan (2014); Watson *et al.* (2015) were introduced as soon as the MinION devices were released, there is still a lack of tools supporting the extraction and analysis of MinION sequence data in real-time.

We developed npReader, an open-source software package to facilitate real-time analysis of Oxford Nanopore MinION sequencing data. npReader is written in Java and hence can run seamlessly on any platforms with minimal dependency. It provides a flexible and comprehensible command line interface, allowing it to be easily incorporated into an analysis pipeline. In addition, it provides a Graphical User Interface (GUI) for users to visually assess the statistics of the run. The package also provides a Java Application Programming Interface (API) allowing developers to extract both based-called sequences and native electrical current data to further develop algorithms for other analyses.

2 METHODS

A MinION device can sequence an array of DNA fragments in parallel. As the DNA fragment translocates through a pore and changes the electrical charge of the pore, the electrical current is captured. The sequence of electrical signal is then translated (base-called) to a possible DNA sequence by the cloud-based Metrichor suite. The various types of data for the fragment, including the native electrical events, event timings and base-called sequence, are stored in a FAST5 format file.

As soon as a DNA fragment is sequenced by the MinION and base-called by the Metrichor cloud, npReader opens the FAST5 file containing the sequence read, extracts the desired information for an initial analysis, and streams the sequence read in FASTA or FASTQ format to one or more real-time analysis pipelines such as that in (Cao *et al.*, 2015). These pipelines can be run on the same computer or on computing clouds. This will extend the range of applications one can analyse in real-time beyond what currently provided by the Metrichor service.

npReader provides a comprehensible and flexible command line interface, thus can be easily integrated into any existing bioinformatics workflow. From the command line, users can specifically select sequence reads that are suitable for the downstream analysis, such as reads with a certain length and quality. The package also contains tools for further binning reads based on their type, length and quality. npReader can also run in GUI mode, enabling the use of pointing devices. Under GUI mode, npReader can concurrently gather and display useful statistics for the current run (See Fig 1). The plots of these statistics are highly customisable and can be exported to an image format ready for publication.

---

*to whom correspondence should be addressed

Fig. 1. A screen shot of npReader. User can specify options of data to be extracted and details of outputs in the left panel. The right panel displays statistics of data, including read counts, read length histogram and read quality in real-time.

npReader is developed in Java. The package includes all the necessary libraries, such as that for statistical analysis and visualisation, hence requires minimal dependency. Apart from the Java Runtime Environment, the only extra dependency is the HDF5 library which is the core for FAST5 format. As a Java application, npReader can run seamlessly cross platforms including Linux, Windows and OSX and hence allows users to run on the laptop connecting to the MinION device or on high performance computing clusters and computing clouds.

Following the recent Oxford Nanopore announcements of their upcoming sequencing platforms, PromethION and GridION, we recognised the need for the development of streaming algorithms to analyse large volumes of sequencing data in real-time. To support this, npReader provides a Java application programming interface to allow researchers to simultaneously retrieve base-called sequences as well as the raw Nanopore event data. The following code snippet demonstrates the extraction of the two dimensional sequence and the event sequence from a FAST5 file within a Java program.

```
import japsa.seq.nanopore.NanoporeReader;
...
NanoporeReader reader
    = new NanoporeReader(fileName);
reader.readData();
reader.close();

// Get 2D read
FastqSequence read2D = reader.getSeq2D();

// Get the squiggle signals
DetectedEvents events = reader.getEvents();
```

3 DISCUSSION

The npReader package provides a simple and yet powerful solution for analysis of MinION sequencing data in real-time. It provides a GUI to allow users to visually examine quality statistics of a run when a DNA sample is being sequenced. Importantly, it can extract the desired information from the run simultaneously and stream into multiple downstream analysis pipelines on high performance computing facilities or on computing clouds, thereby facilitating real-time analysis of the DNA sample. This will allow answering the biological questions at hand in a timely manner, and as soon as sufficient data are obtained.

REFERENCES

Cao, M. D., Ganesamoorthy, D., Elliott, A., Zhang, H., Cooper, M., and Coin, L. J. M. (2015). Real-time strain typing and analysis of antibiotic resistance potential using Nanopore MinION sequencing. *bioRxiv*. DOI:10.1101/019356.

Loman, N. J. and Quinlan, A. R. (2014). Poretools: a toolkit for analyzing nanopore sequence data. *Bioinformatics*, **30**(23), 3399–3401.

Watson, M., Thomson, M., Risse, J., Talbot, R., Santoyo-Lopez, J., Gharbi, K., and Blaxter, M. (2015). poRe: an R package for the visualization and analysis of nanopore sequencing data. *Bioinformatics*, **31**(1), 114–115.