A new method for the production of NaS reads

Pierre Morisse

University of Rouen
pierre.morisse2@univ-rouen.fr

January 20, 2017

Abstract

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Introduction

Since a few years, long reads sequencing technologies are being developed, and allow the solving of assembly problems for large and complex genomes that were impossible with the use of short reads sequencing technologies alone. The two major actors of these long reads sequencing technologies are Pacific Biosciences and Oxford Nanopore, which, with the release of the MinION device, allowed a low-cost and ???? long reads sequencing.

However, even though long reads can reach lengths of tens of kb, they also reach a very high error rate of around 15% for Pacific Biosciences' reads, and up to 30% for Oxford Nanopore reads, the vast majority of these errors being insertions and deletions . Correcting these long reads before using them to solve assembly problems is therefore mandatory.

Many methods are available for short reads correction, but these methods are not applicable to the long reads, on the one hand because of their much higher error rate, and on the other because most of the error correction tools for short reads focus on substitution errors, the dominant error type in Illumina data, whereas insertions and deletions are more common in long reads.

Recently, several methods for long reads correction have been developed. These methods can be divided into two main categories: either the long reads are selfcorrected by aligning them against each other, or either a hybrid strategy is adopted, in which the long reads are corrected with the help of accurate short reads.

Cite existing selfcorrection tools

Cite existing hybrid correction tools

NaS [3], instead of directly correcting the long reads, uses them as templates to produce synthetic long reads, by mapping short reads both on long reads and against each others. A synthetic long read is thus obtained and used as a correction of a given template long read by assembling a subset of short reads related to the said template.

In this paper, we present a new method to produce synthetic long reads, that gets rid of the time consuming step of aligning all the short reads against each other. Instead, we focus on a seed-and-extend approach where we extend and link together the seeds, found by mapping the short reads on the long reads, with perfectly overlapping *k*-mers from the short reads, found with the help of PgSA [2].

Our experiments show that, while producing comparable results both in terms of length and accuracy of the synthetic long reads, our method is several orders of magnitude faster than NaS.

PgSA Overview

PgSA, along with GkA [7] and CGkA [6] are data structures that allow the indexing of a set of reads, in order to answer the following queries, for a given string f :

1. In which reads does f occur?
2. In how many reads does f occur?
3. What are the occurrence positions of f ?
4. What is the number of occurrences of f ?
5. In which reads does f occur only once?
6. In how many reads does f occur only once?
7. What are the occurrence positions of f in the reads where it occurs only once?

In these queries, f can be given either as a sequence of DNA symbols, or as a couple of numbers, representing respectively a read ID, and the start position of f in that read.

As previously mentioned, in order to answer these queries, an index of the reads has to be built. To do so, PgSA first computes the overlaps between the reads, and merges the reads that do overlap, thus obtaining a pseudogenome, shorter than the naive concatenation of the whole reads set. Then, an auxiliary array is built to allow the retrieval of the reads from the original set in the pseudogenome. Each record of this array associates a read ID in the original reads set to a read offset in the pseudogenome, and contains a flag data that brings complementary information about the said read and that will be used for the handling of the requests.

As the reads are overlapped during the pseudogenome computation, and the auxiliary array doesn't record any information about their lengths, PgSA will only allow the indexing and querying of a set of reads of same length. However, unlike its peers GkA and CGkA, PgSA doesn't set the length of f at compilation time, and thus supports querying for multiple lengths of f without any need to recompute the index, which is why we chose this data structure over the two others.

NaS Overview

NaS is a hybrid method for the error correction of long reads. Unlike other methods, instead of directly correcting the long reads, it rather uses them as templates. Short reads are mapped both on these templates long reads and against each other in order to gather different subsets of short reads, each related to one given template. Once a subset of short reads is obtained, contained short reads are assembled and the produced contig is used as a correction for the related template. More precisely, a synthetic long read is produced as follows:

First, the short reads are aligned on the template long read using BLAT [1] (untrue, blat is used for fast mode, and last for sensitive mode, although the NaS paper only mentions blat), in order to find seeds, which are short reads that correctly align with the template. Then, once these seeds are found, all the other short reads are aligned against each other, and similar reads are recruited, with the help of Commet [4]. Finally, the obtained subset of short reads is assembled using Newbler (CITE), and a contig is produced, and used as the correction of the initial template long read.

Usually, a single contig is produced, but in repeated regions, a few bad reads can be recruited and therefore yield erroneous contigs that must not be associated with the template. To address this issue, and produce a single contig, NaS explicitly builds the contig-graph, weighted with the seeds coverage of the contigs. Once the graph is built, the path with the highest total weight is chosen with the Floyd-Warshall algorithm, and contigs along that path are assembled to generate the final synthetic long read. Finally, the consistency of the synthetic read is checked by aligning initial Illuminia short reads and detecting gap of coverage.

The reads recruitment step is the most important(?) step of the method, as it allows to retrieve short reads corresponding to low quality regions of the template long read. However, this step is also the bottleneck of the whole NaS pipeline, as it is responsible for 70% of the total runtime on average.

NaS is able to generate synthetic long reads up to 60kb, that align entirely to the reference genome with no error, and that spans repetitive regions. On average, the accuracy of synthetic long reads yielded by NaS reaches 99.97%, without any significant length drop compared to the input template long reads.

Our method

Our method, like NaS, aims to use erroneous long reads as templates, and produce synthetic long reads from an assembly of short reads, related to the templates. However, our main objective is to get rid of the time consuming step of reads recruiting, that requires the mapping of all the short reads against each other. To do so, we focus on a seed-and-extend approach, where seeds are extended and linked together by perfectly overlapping k-mers from the short reads data, found with the help of PgSA. The workflow of our method is summarized Figure **??**, and detailed below.

Even though short reads are very accurate, as we seek to use their k-mers to compute perfect overlaps and extend the seeds, we need to get rid of as much sequencing errors as we can in this data. We thus correct the short reads with the help of Quorum [5], which provides a good raise of the accuracy in very little time.

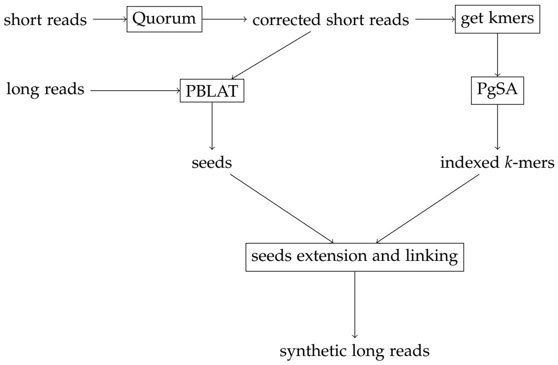

Figure 1: *Our method's workflow*

Once corrected, the k-mers from the short reads and their reverse complements are extracted with Jellyfish (CITE), and indexed with PgSA, before being queried to extend and link the seeds together during the next steps.

Like in NaS, the seeds are found by mapping the corrected long reads on the template long reads with BLAT, or more exactly PBLAT, a slightly modified version of BLAT that allows multithreaded execution.

Then, for each template, the mapped seeds are checked, and those that overlap over more that a certain length, defined as the size of a k-mer minus 1, are merged. Otherwise, if two seeds do overlap, but not over a sufficient length, only the one with the best alignment score is kept.

Once seeds have been found and merged for each template, our method attempts to link together every couple of seeds, by extending the rightmost k-mer of the left seed with perfectly overlapping k-mers from the short reads, until the leftmost k-mer of the right seed is reached. To do so, PgSA's third request, that gives the occurrences positions of a given string, is looped over to find overlaps of length $k-1$ between the currently considered k-mer and the other k-mers from the set of short reads. When such an overlap is found, the current k-mer is extended with the non-overlapping bases of the new found one, which is then considered for the next extension. If no overlap of length $k-1$ is found, then the length is decreased and overlaps are searched again, as PgSA allows requesting for strings of variable lengths. Overlap length thus keeps on decreasing until an overlap is found, or until the minimum length, fixed as $k/2$ is reached.

When requesting PgSA to find overlapping k-mers, it possible to find multiple k-mers that perfectly overlap with the currently considered k-mer. In such cases, all possible extensions are checked with the use of backtracking, to find the one that will allow correct linking of the two seeds. However, to avoid long runtimes and intensive computations, a threshold on the maximum number of backtracks is set. If this threshold, or the previously defined minimum overlap length, is reached and no path has been found to link the two seeds, then the linking is given up, a new linking is computed for the next seeds couple, and a fragmented synthetic long read is produced.

Finally, it is obvious that seeds don't always map right at the beginning and until the end of the templates. Thus, in order to get as close as possible to the original templates' lengths, once all the seeds have been linked, we keep extending the so produced synthetic long read, on the left of the leftmost seed, and on the right of the rightmost seed, until we reach the template's borders, or an ambiguity. This happens when multiple k-mers perfectly overlap the currently considered k-mer, and that its extension is therefore possible with every of these different k-mers. As we have no clue as to which one to chose and to continue the extension with, nor precise destination, as when we attempt to link two seeds, then extension is simply stopped when such a situation is reached.

Results and discussions

We tested our method on various datasets in order to compare it with NaS. Results of these experiments are given Table [it doesn't exist yet].

Conclusions

We developed ...

Competing interests

None declared.

References

[1] W. J. Kent. BLAT âĂŤ The BLAST -Like Alignment Tool. *Genome research*, 12:656–664, 2002.

[2] T. Kowalski, S. Grabowski, and S. Deorowicz. Indexing arbitrary-length k-mers in sequencing reads. *PLoS ONE*, 10(7):1–14, 2015.

[3] M.-A. Madoui, S. Engelen, C. Cruaud, C. Belser, L. Bertrand, A. Alberti, A. Lemainque, P. Wincker, and J.-M. Aury. Genome assembly using Nanopore-guided long and error-free DNA reads. *BMC Genomics*, 16:327, 2015.

[4] N. Maillet, G. Collet, T. Vannier, D. Lavenier, and P. Peterlongo. Commet: Comparing and combining multiple metagenomic datasets. *Proceedings - 2014 IEEE International Conference on Bioinformatics and Biomedicine*, IEEE BIBM(November):94–98, 2014.

[5] G. Marçais, J. A. Yorke, and A. Zimin. QuorUM: An Error Corrector for Illumina Reads. pages 1–13, 2015.

[6] V. Niko. Scalable and Versatile k -mer Indexing for High-Throughput Sequencing Data. (250345):237–248, 2013.

[7] N. Philippe, M. Salson, T. Lecroq, M. Leonard, T. Commes, and E. Rivals. Querying large read collections in main memory: a versatile data structure. *BMC bioinformatics*, 12(1):242, 2011.