HG-CoLoR: A new method for the production of synthetic long reads

Pierre MORISSE, Thierry LECROQ and Arnaud LEFEBVRE
Normandie-Université, UNIROUEN, LITIS EA4108, 76821, Mt-St-Aignan, France

Corresponding author: pierre.morisse2@univ-rouen.fr

Abstract *The recent rise of long read sequencing technologies allows the solving of assembly problems for large and complex genomes that were, until then, unsolvable with the use of short read sequencing technologies alone. Despite the fact that they can reach lengths of tens of kbps, these long reads are very noisy, and can reach an error rate as high as 30%, involving mandatory error correction before using them to efficiently solve assembly problems. However, as the vast majority of these errors are insertions and deletions, classical error correction tools developed for short reads, which mainly focus on substitution errors, are not effective for correcting long reads. Therefore, several new methods specifically designed for long read error correction have recently been developed. In particular, NaS, instead of directly correcting the long reads, proposes to use them as templates in order to produce synthetic long reads from assemblies of related accurate short reads as corrections. Following this idea, we introduce HG-CoLoR (Hybrid Graph for the error Correction of Long Reads), a new tool for the production of synthetic long reads, that gets rid of the need to align the short reads against each other, which is the bottleneck from NaS. Indeed, HG-CoLoR focuses on a seed-and-extend approach based on a hybrid graph built from the short reads. Our experiments show that, while producing comparable results both in terms of length and accuracy of the synthetic long reads, HG-CoLoR is several times faster than NaS, and also yields better assembly results than other state-of-the-art long read hybrid error correction methods. HG-CoLoR is available from https://github.com/pierre-morisse/HG-CoLoR.*

Keywords NGS, long reads, correction, assembly

1 Introduction

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

However, even though long reads can reach lengths of tens of kbps, they also reach a very high error rate of around 15% for Pacific Biosciences, and up to 30% for Oxford Nanopore, the vast majority of these errors being insertions and deletions. Due to this high error rate, correcting these long reads before using them to efficiently solve assembly problems is mandatory. Many methods are available for short read correction, but these methods are not applicable to long reads, on the one hand because of their much higher error rate, and on the other hand, 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 read 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 (HGAP [1], PBcR [2]), or either a hybrid strategy is adopted, in which the long reads are corrected with the help of accurate short reads (LSC [3], proovread [4], CoLoRMap [5]). de Bruijn graph [6] based methods, where the long reads are mapped on the graph, and erroneous regions corrected by traversing its paths, also started to develop recently, in the hybrid case (LoRDEC [7], Jabba [8]), as well as in the non-hybrid case (LoRMA [9]).

NaS [10], instead of directly correcting the long reads, uses them as templates to produce synthetic long reads from assemblies of related accurate short reads. The short reads are mapped both on these templates, and against each other, in order to associate a subset of short reads to each template. A synthetic long read is thus obtained and used as the correction of a given template by assembling the subset of short reads associated to it.

In this paper, we introduce HG-CoLoR, a new long read hybrid error correction method that combines both the main idea from NaS to produce synthetic long reads, and the use of a graph, in order to get rid of

the time consuming step of aligning all the short reads against each other. HG-CoLoR indeed focuses on a seed-and-extend approach where the seeds, which are short reads that align correctly on the long reads, are used as anchor points on a graph that is traversed in order to link them together and to produce the synthetic long reads. This graph, which is simulated with the help of PgSA [11], is actually a hybrid structure between a de Bruijn graph and an overlap graph [12], is built from the short reads' k-mers, and allows to compute perfect overlaps of variable length between these k-mers.

Our experiments show that, while producing comparable results both in terms of length and accuracy of the synthetic long reads, HG-CoLoR is several times faster than NaS, and also yields better assembly results than other state-of-the-art long read hybrid error correction methods.

For the sake of understanding, we first give an overview of NaS, and describe our hybrid graph and the way it is simulated, before introducing HG-CoLoR.

2 NaS Overview

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

First, the short reads are aligned on the template using BLAT [13] in fast mode, or LAST [14] in sensitive mode, in order to find seeds, which are short reads that align correctly on the template. Then, once these seeds have been found, all the short reads are aligned against each other, and similar reads, which are reads that share a certain number of non-overlapping k-mers with the seeds, are recruited with the help of Commet [15]. Finally, the obtained subset of short reads is assembled using Newbler (unpublished), and a contig is produced, and used as the correction of the initial template

The reads recruitment is the most crucial step of the method, as it allows to retrieve short reads corresponding to low quality regions of the template. 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 60 kbps, that align entirely on the reference genome and that span repetitive regions. On average, the accuracy of the synthetic long reads produced by NaS reaches 99.75%, without any significant length drop compared to the input long reads. Moreover, these synthetic long reads also yield highly contiguous assembly results, and thus provide an interesting alternative to classical long read hybrid error correction.

3 Hybrid graph

As previously mentioned, the graph used by HG-CoLoR is a hybrid structure between a de Bruijn graph and an overlap graph. This hybrid graph is simulated with the help of PgSA, which is a data structure that allows the indexing of a set of reads of constant length, in order to answer different queries, for a given string f. For place sake, we do not detail how the index is built, the complete list of queries, nor how they are processed. For more details, one can refer to [11]. We simply mention that PgSA supports querying for variable lengths of f without recomputing the index, and that one of the queries returns the positions of all the occurrences of f in the different reads of the set.

This way, using PgSA to index a set of reads, and looping over the aforementioned query, allows to compute perfect overlaps of variable length between the reads, thus simulating an overlap graph. In the same fashion, indexing the k-mers from a set of reads, and looping over the aforementioned query, fixing the length of the queries strings as $k-1$, allows to compute perfect overlaps of length $k-1$ between the k-mers, thus simulating a de Bruijn graph. However, indexing the k-mers from a set of reads, and looping over the aforementioned query, of course, also allows to compute perfect overlaps of variable length between the different k-mers, thus simulating a hybrid structure between a de Bruijn graph and an overlap graph. To the best of our knowledge, this is the first time such a structure is mentioned. For better understanding, an example of a simple graph is given in Figure 1.

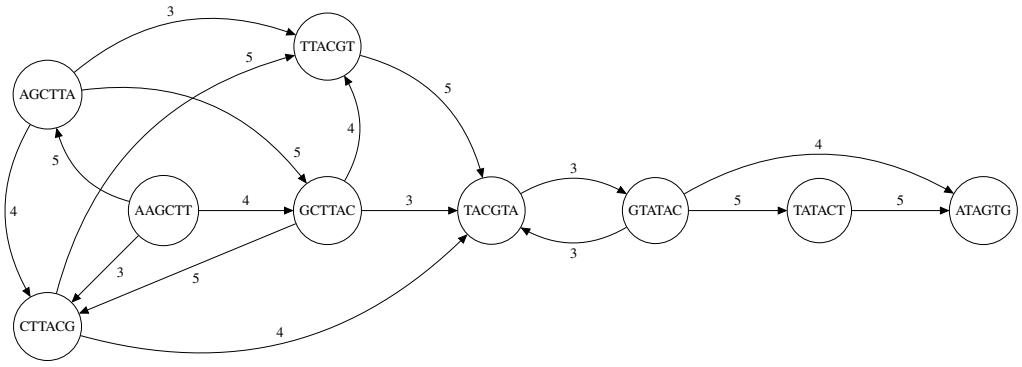

Fig. 1. A simple example of our graph, when fixing the length of the k-mers to 6, computing overlaps of minimum length 3, and building from the three following reads: AAGCTTAC, CTTACGTA, GTATACTG. Numbers on the edges of the graph represent the overlap length between the k-mers.

4 HG-CoLoR description

HG-CoLoR, like NaS, aims to use erroneous long reads as templates, and to produce synthetic long reads from assemblies of short reads related to these templates. However, its 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, it focuses on a seed-and-extend approach where the seeds are found in the same way as NaS, and where the k-mers from the short reads, and their reverse-complements, are indexed with PgSA, to simulate the previously described graph. This graph is then traversed, in order to extend and link together the seeds, used as anchor points, by directly assembling the short reads' k-mers during the traversal. HG-CoLoR's workflow is summarized in Figure 2, and its four main steps are described below.

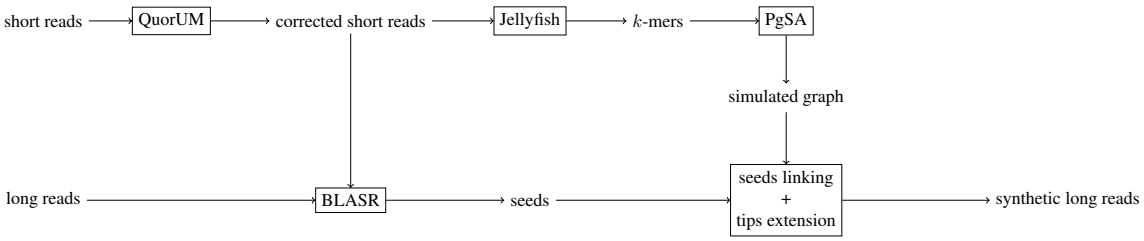

Fig. 2. HG-CoLoR's workflow. First, the short reads are corrected in order to get rid of as much sequencing errors as possible. Then, all the k-mers from the corrected short reads, and their reverse-complements, are obtained with Jellyfish, and indexed with PgSA, to simulate the graph. The corrected short reads are aligned on the long reads with BLASR to find seeds, and each long read is then considered as a template, and processed independently. For a given template, the graph is traversed in order to extend and link together the associated seeds, used as anchor points. Then, the tips of the sequence obtained after the seeds linking step are extended in both directions by traversing the graph, to reach the initial template's borders. Finally, the synthetic long read is output.

4.1 Short reads correction and indexing

Even though short reads are very accurate prior to any correction, as HG-CoLoR seeks to use their k-mers to simulate a graph, and traverse it to extend and link the seeds together, it needs to get rid of as much sequencing errors as it can in this data. Thus, prior to any other step, the short reads are corrected with the help of QuorUM [16], which is able to provide a good raise of the accuracy in very little time. Then, the k-mers from the corrected short reads, and their reverse-complements, are extracted with Jellyfish [17], and indexed with PgSA, in order to simulate the graph that will be traversed during the following steps.

4.2 Seeds retrieving and merging

Like with NaS, the seeds are found by mapping the corrected short reads on the long reads, used as templates. This is done with the help of BLASR [18], an alignment tool specifically designed to align long reads dominated by insertion and deletion errors. Then, each template is processed independently, and two phases of analyze and merging are applied to the associated seeds. First, if the mapping positions of a given couple of

seeds imply that they overlap on the template over a sufficient length, their assumed overlapping sequences are compared, and the two seeds are merged accordingly. If the mapping positions indicate that the two seeds do overlap on the template, but not over a sufficient length, or if the assumed overlapping sequences do not coincide, only the seed with the best alignment score is kept. Then, once all the seeds with overlapping mapping positions have been merged or filtered out, sequence overlaps between consecutive seeds are computed. As in the previous step, if a given seed overlaps the following one over a sufficient length, the two seeds are merged.

4.3 Seeds linking

Once the seeds have been found and merged for all of the templates, HG-CoLoR once again processes each template independently and attempts to link the related seeds together by considering them as couples, and traversing the graph. The rightmost k-mer of the left seed (source) and the leftmost k-mer of the right seed (destination) are used as anchor points, and the source is extended with perfectly overlapping k-mers from the corrected short reads, found by following the paths of the graph, until the destination is reached. When facing branching paths, every possible path is explored with the use of backtracking, to find the one that will allow correct linking of the source to destination. Of course, HG-CoLoR explores these different paths in decreasing order of the overlaps lengths, which means that edges representing longer overlaps are always explored before those representing shorter ones. It also only explores edges that represent overlaps that are longer than a defined minimum length. Moreover, as short reads from a different region of the reference genome can align on the template and can be used as seeds, thus leading to impossible linkings, a threshold on the maximum number of backtracks is set, to avoid useless important runtime and intensive computation.

If this threshold is reached, and no path has been found to link the source to the destination, the current linking iteration is given up. When such a situation occurs, two different cases have to be taken into account. In the first case, if no seeds have been linked so far, the current source is simply ignored, and a new linking iteration is computed for the next couple of seeds. In the second case, if seeds have already been linked previously, the source remains the same, the destination seed that could not be reached is ignored, and the destination is defined as the next seed for the next linking iteration. An illustration of these two different cases is given in Figure 3.

However, in the second case, as this process of skipping a seed in the middle of the template can provoke an important number of failed linking attempts, if seeds from a wrong region are present in great proportion on the template, a threshold on the maximum number of seeds that can be skipped is set. Once this threshold is reached, if the sequence obtained from the previously linked seeds could not be extended to reach one of the remaining seeds, HG-CoLoR attempts to produce a fragmented synthetic long read: the part corresponding to the seeds linked so far is output, and the graph is traversed again, in order to try to link the remaining seeds together, independently of the previous part.

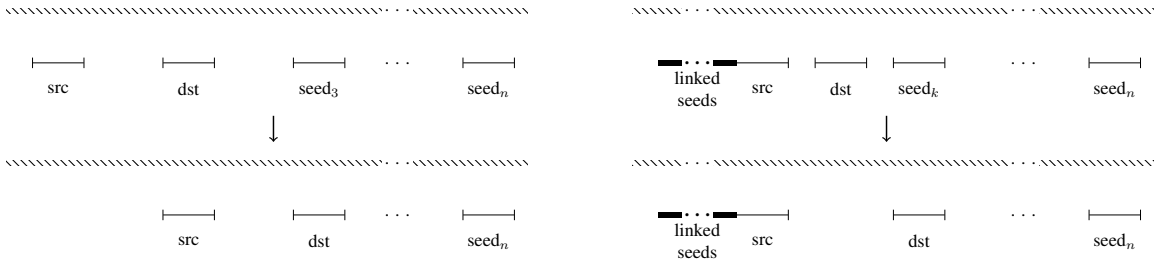

Fig. 3. Illustration of the different cases of seed skips. Hatched lines represent the templates, standard segments represent the seeds, and bold segments represent the sequences obtained from the previously linked seeds. First case (left): No seeds have been linked so far, the current source seed is simply ignored, and both the source and the destination are moved to the next couple of seeds. Second case (right): Seeds have already been linked previously, the source remains the same, the destination seed that could not be reached is ignored, and the destination is defined as the next seed.

4.4 Tips extension

Finally, it is obvious that the seeds do not 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 of a given template have been linked, HG-CoLoR keeps on traversing the graph and extending the tips of the produced synthetic long read, on the left of the leftmost seed, and on the right of the rightmost seed, until they reach the template's borders, or a branching path. Indeed, in the case of tips extension, when facing a branching

path, HG-CoLoR has no clue as to which path to chose and continue the extension with, nor any anchor points, unlike when it attempts to link two seeds together. Therefore, backtracking is useless and the extension is simply stopped when such a situation occurs. In the case of fragmented synthetic long reads, as HG-CoLoR can not properly rely on the template's borders, every fragment is extended until a branching path is reached.

5 Results and discussion

We compare the quality of our synthetic long reads with those produced by NaS, and also with the corrected long reads produced by two other state-of-the-art hybrid error correction methods, namely CoLoRMap and Jabba. We compare the results both in terms of alignment identity of the corrected reads, and in terms of quality of the assemblies that could be generated from these reads.

5.1 Parameters

We ran multiple rounds of correction with HG-CoLoR on the different datasets to experiment with the parameters, and find the combination that would produce the best results. Thereby, we found that a k-mer value of 64 for the graph construction yielded the best compromise between identity, genome coverage, and average length of the output synthetic long reads. The minimum overlap length to allow the merging of two seeds during the second step was set to 63, accordingly to the k-mer size chosen for the graph construction. The minimum overlap length allowed to explore an edge during the graph traversal was set to 59, as decreasing it more yielded unsatisfying results, and increasing it would make our graph closer to an actual de Bruijn graph than to the hybrid graph it's supposed to be. The maximum number of backtracks was set to 1,125, as decreasing it more drastically impacted the quality of the produced synthetic long reads, and increasing it, even to very large values, barely yielded better results, but greatly increased the runtime. For the same reason, the maximum number of seed skips was set to 5. For the mapping of the short reads on the long reads, BLASR was used with default parameters except for bestn, that was set to 30 instead of 10. Yet again, increasing this parameter to larger values only impacted the runtime, and did not improve the correction results enough to be interesting, while decreasing it induced a drop of the number of output synthetic long reads. Finally, GNU Parallel [19] was used to allow HG-CoLoR to run on multiple processes. CoLoRMap was run with default parameters. Following the authors' recommendations, before running Jabba, the short reads were corrected with Karect [20], and the de Bruijn graph was constructed and corrected with Brownie (unpublished). A k-mer size of 75 was then chosen for the graph construction. All tools were run with 16 processes.

5.2 Datasets

As we mainly seek to compare our results with NaS, we use the same data to allow a better comparison. This data is composed of both long Oxford Nanopore reads and short Illumina reads for three different genomes: *Acinetobacter baylyi*, *Escherichia coli*, and *Saccharomyces cerevisae*. Details are given in Table 1.

Dataset	Reference genome				Oxford Nanopore data			Illumina data		
	Name	Strain	Reference sequence	Genome size	# Reads	Average length	Coverage	# Reads	Read length	Coverage
A. baylyi	*A. baylyi*	ADP1	CR543861	3.6 Mbp	89,011	4,284	44x	900,000	250	50x
E. coli	*E. coli*	K-12 substr. MG1655	NC_000913	4.6 Mbp	22,270	5,999	28x	775,500	300	50x
Yeast	*S. cerevisae*	W303	scf7180000000084-113	12.4 Mbp	205,923	5,698	31x	2,500,000	250	50x

Tab. 1. Description of the datasets used in our experiments. Both MinION and Illumina data are available from the Genoscope's website http://www.genoscope.cns.fr/externe/nas/datasets.html.

5.3 Alignment-based comparison

The previously described long reads datasets were aligned with Last prior to any correction. The four different correction tools were then applied, and the obtained corrected long reads were aligned with BWA mem [21]. Results are given in Table 2 and discussed below.

We notice that, unlike the other methods, CoLoRMap output all the long reads and not only the ones it managed to correct. As the reads that could be corrected were not tagged in any way and could therefore not be extracted, it appears that CoLoRMap performed the worst correction, and did not manage to improve the accuracy of the long reads at all, except for the *E. coli* dataset. These poor results are probably due to the fact that only a few reads could be corrected, as CoLoRMap is designed to correct long reads from Pacific Biosciences, that have an error rate of about 15%, whereas the long reads used in our experiments were from

Dataset	Method	# Reads	Average length	Cumulatize size	# Aligned reads	Average identity	# Error-free reads	Genome coverage	Runtime
A. baylyi	Original	89,011	4,284	381,365,755	29,954 (33.65%)	70.09%	0 (0%)	100%	N.A.
	CoLoRMap	89,011	4,355	387,609,994	18,085 (20.32%)	67.93%	2 (0.01%)	100%	14h33min
	Jabba	17,476	10,260	179,309,738	17,476 (100%)	99.40%	16,893 (96.66%)	99.80%	12min30
	NaS (fast)	24,063	8,840	212,707,189	24,063 (100%)	99.82%	22,984 (95.52%)	100%	-
	NaS (sensitive)	28,492	9,530	271,526,778	28,492 (100%)	99.83%	27,190 (95.43%)	100%	-
	HG-CoLoR	23,465	11,137	261,327,970	23,461 (99.98%)	99.44%	20,906 (89.11%)	100 %	21h08min
E. coli	Original	22,270	5,999	133,607,392	22,170 (99.55%)	79.46%	0 (0%)	100%	N.A.
	CoLoRMap	22,270	6,219	138,489,144	21,784 (97.82%)	89.02%	152 (0.70%)	100%	8h26min
	Jabba	22,065	5,794	127,848,525	22,065 (100%)	99.81%	21,850 (99.03%)	99.41%	12min56
	NaS (fast)	21,818	7,926	172,918,739	21,818 (100%)	99.86%	20,383 (93.42%)	100%	-
	NaS (sensitive)	22,144	8,307	183,958,832	22,144 (100%)	99.86%	20,627 (93.15%)	100%	-
	HG-CoLoR	22,549	5,897	132,979,813	22,549 (100%)	99.59%	19,676 (87.26%)	100%	15h15min
Yeast	Original	205,923	5,698	1,173,389,509	68,215 (33.13%)	55.49%	0 (0%)	99.90%	N.A.
	CoLoRMap	205,923	5,737	1,181,298,941	40,530 (19.68%)	39.93%	23 (0.06%)	99.40%	37h36min
	Jabba	36,958	6,613	244,402,749	36,855 (99.72%)	99.55%	34,028 (92.33%)	93.21%	44min05
	NaS (fast)	71,793	5,938	426,326,355	71,664 (99.82%)	99.59%	59,788 (83.43%)	98.70%	-
	NaS (sensitive)	85,432	6,770	578,351,588	85,288 (99.83%)	99.53%	69,816 (81.86%)	99.17%	-
	HG-CoLoR	71,284	6,576	468,735,999	71,161 (99.83%)	99.18%	55,240 (77.63%)	98.39%	112h20min

Tab. 2. Runtime and statistics of the long reads, before and after correction by the different tools. NaS runtimes are omitted because results did not compute in 3 days, even for *E. coli* in fast mode. NaS reads were therefore obtained from the Genoscope's website to allow comparison.

Oxford Nanopore, and reached an error of at least 30% for the two other datasets.

Jabba clearly performed the best when it comes to runtime, outperforming all the other tools by several orders of magnitude. It also produced corrected long reads that aligned with a high identity, a great proportion of them aligning with no error. However, although highly accurate, these corrected long reads did not manage to completely cover any of the studied reference genomes.

When it comes to this point, only NaS and HG-CoLoR managed to cover the whole reference genomes with high identity, except for Yeast, due to the fact that even the original long reads did not cover the whole genome. This proves that focusing on the production of synthetic long reads is indeed a good alternative to classical long read hybrid error correction. Moreover, HG-CoLoR outperforming Jabba in terms of genome coverage also clearly underlines the usefulness of our hybrid graph, showing that it allows to resolve the different regions of the reference genomes better than a classical de Bruijn graph.

On the three datasets, NaS yielded more synthetic long reads than HG-CoLoR, both in fast and sensitive mode, the slight advantage of HG-CoLoR on the *E. coli* dataset coming from the production of fragmented synthetic long reads, rather than from a greater number of processed templates. In both modes, the synthetic long reads produced by NaS also aligned with a slightly higher identity than those produce by HG-CoLoR, and a greater proportion was therefore error-free. As for the average length and the cumulative size of the synthetic long reads, HG-CoLoR performances were highly similar to NaS's, except on the *E.coli* dataset, where the advantage of NaS is probably due to the high quality of the original templates, and to the fact that it can recruit short reads outside of the templates, while HG-CoLoR stops once the borders are reached. However, despite its slight disadvantage on the aforementioned metrics, HG-CoLoR was at least four times faster than NaS, even in fast mode.

5.4 Assembly-based comparison

All the corrected long reads datasets previously described were assembled using Canu [22], without the correction and trimming steps. The following parameters were used for the assembly of all the datasets: OvlMerSize=17, MhapMerSize=17, OvlMerDistinct=0.9925, OvlMerTotal=0.9925. The correctedErrorRate parameter was tuned independently for each dataset. It was set at 0.07 for *A. baylyi*, at 0.085 for *E. coli* and at 0.125 for Yeast. Results are given in Table 3 and discussed below.

In agreement with what we observed in Table 2, the low accuracy of the long reads corrected by CoLoRMap resulted in impossible assemblies. Only the corrected long reads of the *E. coli* dataset could be assembled, due to their original high accuracy, but the generated assembly did not cover the whole genome, and displayed the worst identity among all the other assemblies.

As for Jabba, the fact that the corrected long reads did not manage to cover the whole reference genomes resulted in highly fragmented assemblies, that could not resolve large regions of the reference genomes. As a result, long reads corrected by Jabba yielded the least covering assemblies, despite their high average length

Dataset	Method	# Reads	# Expected contigs	# Obtained contigs	Genome coverage	Identity
A. baylyi	CoLoRMap	89,011	1	-	-	-
	Jabba	17,476	1	13	89.43%	99.93%
	NaS (fast)	24,063	1	1	100 %	99.99 %
	NaS (sensitive)	28,492	1	2	99.72%	99.98%
	HG-CoLoR	23,465	1	1	99.97%	99.93%
E. coli	CoLoRMap	22,270	1	29	97,74%	99.81%
	Jabba	22,065	1	41	95.76%	99.92%
	NaS (fast)	21,818	1	1	99.90 %	99.99%
	NaS (sensitive)	22,144	1	2	100%	99.99%
	HG-CoLoR	22,549	1	2	99.95%	99.95%
Yeast	CoLoRMap	205,923	30	-	-	-
	Jabba	36,958	30	134	70.52%	99.83%
	NaS (fast)	71,793	30	123	97.44%	99.77%
	NaS (sensitive)	85,432	30	123	96.98%	99.80%
	HG-CoLoR	71,284	30	109	92.76%	99.63%

Tab. 3. Statistics of the assemblies that were generated from the long reads, after correction by the different tools. CoLoRMap results are ommitted for the *A. baylyi* and Yeast datasets, because Canu did not manage to assemble the sets of corrected reads.

and high accuracy. This underlines the fact that, although it is extremely fast, Jabba does not seem to be adapted for correcting long reads prior to an assembly.

Surprisingly, for all the datasets, the sensitive mode of NaS produced synthetic long reads that resulted in slightly less satisfying assemblies than the fast mode. However, the difference was not significant, and adapting the parameters of Canu to match the synthetic long reads produced in sensitive mode addressed this issue.

Therefore, only the synthetic long reads produced by NaS and HG-CoLoR could be assembled into a decent number of contigs, covering the reference genomes well, and with a high identity. However, for the Yeast dataset, none of these two tools managed to produce synthetic long reads allowing to get close to the expected number of contigs, nor to the full genome coverage. This is probably due to the fact that the original long reads were of really poor quality, displaying an error rate of almost 45%, and did not cover the whole genome. They were indeed sequenced with an old chemistry, and it is more than likely that, with long reads from a more recent one as templates, both NaS and HG-CoLoR could produce synthetic long reads that would greatly reduce the number of contigs and increase the genome coverage of the assembly. We can also suppose that NaS outperforms HG-CoLoR on this dataset for this very same reason, and that using long reads from a more recent chemistry as templates would allow HG-CoLoR to compare much better to NaS.

Once again, these results prove that focusing on the production of synthetic long reads, rather than on the direct correction of the long reads, is a good alternative to classical long read hybrid error correction.

6 Conclusion

We described HG-CoLoR, a new hybrid method for the error correction of long reads, that, like NaS, uses long reads as templates and focuses on the production of synthetic long reads, rather than on the direct correction of the input long reads. Our method, instead of aligning the short reads against each other in a recruiting step, like NaS, focuses on a seed-and-extend approach and introduces a brand new idea of using a hybrid structure between a de Bruijn graph and an overlap graph. This graph, which is built from the short reads' k-mers, and simulated with PgSA, is used to extend and link together the seeds, which are short reads that align correctly on the input long reads, by a simple traversal, using them as anchor points. Therefore, the synthetic long reads are produced by directly assembling the short reads' k-mers during the traversal, without using any other proper assembly tool.

We tested this new method and compared it with NaS, CoLoRMap and Jabba on Oxford Nanopore long reads from three different genomes, namely *A. baylyi*, *E. coli*, and *S. cerevisae*. On these three datasets, HG-CoLoR yielded results that compared well with NaS, while being several times faster, CoLoRMap produced corrected reads of poor quality, and Jabba, while being the fastest tool, produced accurate corrected reads that however did not cover the whole reference genomes. As a result, only the synthetic long reads produced by NaS and HG-CoLoR could be assembled into a decent number of contigs, covering well the reference genomes, although NaS outperformed HG-CoLoR on the *S. cerevisae* dataset.

The development of this method shows that, when having anchor points, the previously introduced hybrid graph can prove useful for hybrid error correction of long reads, and can even yield better results than a classical de Bruijn graph. For future works, it could be interesting to focus more on this graph, and directly build it instead of simulating it, in order to directly map the long reads on the graph, like Jabba, thus skipping the alignment step of the short reads on the long reads, and reducing the runtime.

Acknowledgements

The authors would like to thank the Genoscope team for the availability of all the data used in this paper.

References

[1] Chen-Shan Chin, David H Alexander, Patrick Marks, Aaron A Klammer, James Drake, Cheryl Heiner, Alicia Clum, Alex Copeland, John Huddleston, Evan E Eichler, Stephen W Turner, and Jonas Korlach. Nonhybrid, finished microbial genome assemblies from long-read SMRT sequencing data. *Nature Methods*, 10(6):563–569, 2013.

[2] Konstantin Berlin, Sergey Koren, Chen-Shan Chin, James P Drake, Jane M Landolin, and Adam M Phillippy. Assembling large genomes with single-molecule sequencing and locality-sensitive hashing. *Nature biotechnology*, 33(6):623–630, 2015.

[3] Kin Fai Au, Jason G. Underwood, Lawrence Lee, and Wing Hung Wong. Improving PacBio Long Read Accuracy by Short Read Alignment. *PLoS ONE*, 7(10):1–8, 2012.

[4] Thomas Hackl, Rainer Hedrich, Jörg Schultz, and Frank Förster. Proovread: Large-scale high-accuracy PacBio correction through iterative short read consensus. *Bioinformatics*, 30(21):3004–3011, 2014.

[5] Ehsan Haghshenas, Faraz Hach, S Cenk Sahinalp, and Cedric Chauve. CoLoRMap: Correcting Long Reads by Mapping short reads. *Bioinformatics*, 32(17):i545–i551, 2016.

[6] Nicolaas Govert de Bruijn. A combinatorial problem. *Koninklijke Nederlandse Akademie van Wetenschappen te Amsterdam*, 49(7):758–764, 1946.

[7] Leena Salmela and Eric Rivals. LoRDEC: Accurate and efficient long read error correction. *Bioinformatics*, 30(24):3506–3514, 2014.

[8] Giles Miclotte, Mahdi Heydari, Piet Demeester, Stephane Rombauts, Yves Van de Peer, Pieter Audenaert, and Jan Fostier. Jabba: hybrid error correction for long sequencing reads. *Algorithms Mol Biol*, 11:10, 2016.

[9] Leena Salmela, Riku Walve, Eric Rivals, and Esko Ukkonen. Accurate selfcorrection of errors in long reads using de Bruijn graphs. *Bioinformatics*, 33(6):799–806, 2017.

[10] Mohammed-Amin Madoui, Stefan Engelen, Corinne Cruaud, Caroline Belser, Laurie Bertrand, Adriana Alberti, Arnaud Lemainque, Patrick Wincker, and Jean-Marc Aury. Genome assembly using Nanopore-guided long and error-free DNA reads. *BMC Genomics*, 16:327, 2015.

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

[12] Andrzej Ehrenfeucht, Tero Harju, Ion Petre, David M Prescott, and Grzegorz Rozenberg. *Overlap Graphs*, pages 99–108. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[13] W James Kent. BLAT — The BLAST -Like Alignment Tool. *Genome research*, 12:656–664, 2002.

[14] Szymon M Kielbasa, Raymond Wan, Kengo Sato, Szymon M Kiebasa, Paul Horton, and Martin C Frith. Adaptive seeds tame genomic sequence comparison. *Genome Research*, 21(3):487–493, 2011.

[15] Nicolas Maillet, Guillaume Collet, Thomas Vannier, Dominique Lavenier, and Pierre Peterlongo. Commet: Comparing and combining multiple metagenomic datasets. In *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2014.

[16] Guillaume Marçais, James A Yorke, and Aleksey Zimin. QuorUM: An Error Corrector for Illumina Reads. *PLOS ONE*, 10(6):1–13, 2015.

[17] Guillaume Marçais and Carl Kingsford. A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. *Bioinformatics*, 27(6):764–770, 2011.

[18] Mark J Chaisson and Glenn Tesler. Mapping single molecule sequencing reads using basic local alignment with successive refinement (BLASR): application and theory. *BMC bioinformatics*, 13(1):238, 2012.

[19] Ole Tange. GNU Parallel - The Command-Line Power Tool. *;login: The USENIX Magazine*, 36(1):42–47, 2011.

[20] Amin Allam, Panos Kalnis, and Victor Solovyev. Karect: accurate correction of substitution, insertion and deletion errors for next-generation sequencing data. *Bioinformatics*, 31(21):3421–3428, 2015.

[21] Heng Li and Richard Durbin. Fast and accurate long-read alignment with Burrows-Wheeler transform. *Bioinformatics*, 26(5):589–595, 2010.

[22] Sergey Koren, Brian P. Walenz, Konstantin Berlin, Jason R. Miller, and Adam M. Phillippy. Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation. *bioRxiv*, 2016.