MTG-Link: filling gaps in draft genome assemblies with linked read data

Anne GUICHARD[1,2], Fabrice LEGEAI[1,2], Denis TAGU[1] and Claire LEMAITRE[2]
[1] INRAE, Agrocampus Ouest, Université de Rennes, IGEPP, F-35650 Le Rheu, France
[2] Univ Rennes, CNRS, Inria, IRISA - UMR 6074, F-35000 Rennes, France

Corresponding author: anne.guichard@irisa.fr

Abstract *De novo genome assembly is a challenging task, especially for large non-model organism genomes. Low sequence coverage, genomic repeats and heterozygosity often create ambiguities in the assembly, and result in undefined sequences between contigs called "gaps". Hence, filling gaps in draft genomes has become a natural sub-problem of many de novo genome assembly projects. Even though there are several tools for closing gaps, to our knowledge none uses the long-range information of the linked read data. Linked read technologies have a great potential for filling gaps in draft genomes as they provide long-range information while maintaining the power and accuracy of short-read sequencing. In this work, we present MTG-Link, a novel gap-filling tool dedicated to linked read data. Taking advantage of the barcode information contained in the linked read dataset, a subsample of reads is first selected for each gap. These reads are then locally assembled and the resulting gap-filled sequences are automatically evaluated. We validated our approach on a real 10X genomics linked read dataset, on a set of simulated gaps, and showed that the read subsampling step of MTG-Link enables to get better gap assemblies in a time/memory efficient manner. We also applied MTG-Link on individual genomes of a mimetic butterfly (Heliconius numata), where it significantly improved the contiguity of a 1.3 Mb locus of biological interest.*

MTG-Link is freely available at https://github.com/anne-gcd/MTG-Link.

Keywords High throughput sequencing, Genome assembly, Gap-filling, Linked reads

1 Introduction

The fast development of both second and third generation sequencing technologies have been accompanied by an increased growth of the number of de novo genome assemblies, with better quality. Complete genome assemblies are crucial for downstream analysis as they enable to get better genome annotations, less genotyping errors and provide valuable information on structural variations [1].

Long-read sequencing technologies such as Pacific Biosciences and Oxford Nanopore are expected to greatly improve the quality of the assembled draft genomes. Indeed, these technologies offer much longer reads than short-read sequencing technologies (10-200 kb vs. 100-250 bp), giving the ability to span repetitive regions, define haplotypes and resolve structural rearrangements [2,3]. However, relative to short-read sequencing, long-read sequencing suffers from high error rates (10-15% vs. $\leq$ 0.3%) [4] and lower throughput [5]. Synthetic long-read sequencing approaches can also be used for genome assembly, as they provide all the benefits of short-read sequencing, besides incorporating information from long strands of DNA [6]. These include linked reads, which can be employed in synergy with true long reads to get accurate and complete genome assemblies.

With linked read technologies, such as the 10X Genomics Chromium platform, every short reads that have been sequenced from the same long DNA molecule (around 30-50 Kb) are tagged with a specific molecular barcode. Non-contiguous reads sharing the same barcode are referred to as linked reads. By linking the short reads together via a shared barcode, linked read technology provides long-range information while maintaining the power and accuracy of short-read sequencing [7,8]. Low-cost, low-input and high-accuracy linked read technologies have many applications: de novo genome assembly [8], haplotype identification [9] and structural variant calling [10]. The 10x Chromium Genomics company, which popularized this technology [9], recently stopped producing such data.

However, large volumes of data were produced and still need to be properly analyzed, and other linked read technologies such as TELL-Seq [11] and Haplotagging [12] emerged.

Complete and accurate reconstruction of large non-model organism genomes remains challenging with the current technologies and assembly tools. Problems generally reside at regions that are highly repetitive, highly heterozygous or have low coverage. All these features create ambiguities in the overlap detection between reads, resulting in undefined sequences between contigs of unknown or estimated lengths, called *gaps*.

Gap-filling methods aim at recovering the gap sequence between contigs, by performing a local assembly of the sequencing reads between the flanking sequences. Several tools have been developed for local assembly or gap-filling with short read data, such as GapCloser [13], Sealer [14], GapFiller [15], GAPPadder [16] and MindTheGap [17]. Implemented algorithms are quite different: some rely on De Bruijn graphs, others on iterative extensions based on read overlaps. While some methods use the whole input read set for assembly, others select reads of interest based on mate anchoring of paired-end or mate pair reads. Therefore, the former have difficulty assembling repeat-rich gaps while the latter are limited in the gap size. Even though there are several tools for closing gaps with short read data, to our knowledge, there is currently no tool that uses the long-range information of the linked read data, although this type of information has proven to be very useful for assembly issues.

In this work, we present MTG-Link, a novel gap-filling tool for draft genome assemblies dedicated to linked read data. The main feature of MTG-Link is that it takes advantage of the linked-read barcode information to get a subsample of reads of interest for the local assembly of each gap. It also automatically tests different parameters values and performs a qualitative evaluation of the obtained solutions. We validated our approach on a real 10X genomics dataset, in which gaps were simulated, and compared it to MindTheGap, that does not use the barcode information. We showed that the read subsampling step of MTG-Link enables to get better gap assemblies in less CPU time. We then applied our tool on several individual genomes of a mimetic butterfly (*Heliconius numata*) to improve the contiguity of a 1.3 Mb locus of biological interest.

2 Materials and Methods

2.1 Gap-filling with linked read data

Pipeline overview We propose a method, called MTG-Link, that aims at filling gaps in draft genome assemblies using linked read data. The method takes as input a set of linked reads, a GFA file with gap coordinates and an indexed BAM file obtained after mapping the linked reads onto the draft assembly. It outputs the set of gap-filled sequences in FASTA format, as well as an assembly graph file in GFA format, containing the original contigs and the obtained gap-filled sequences of each gap, together with their overlapping relationships.

The method described in this work relies on a three-step pipeline, where each gap is processed independently from the others. The first step uses the barcode information of the linked read dataset to get a subsample of reads of potential interest for gap-filling. The second step performs local assembly using this subsample of linked reads. Two different assembly algorithms are implemented and can be interchangeably used. The first one, called hereafter the *De Bruijn Graph (DBG) algorithm*, uses a de Bruijn graph data structure, and the second one, called the *Iterative Read Overlap algorithm*, is based on on-the-fly computations of read overlaps. The third step evaluates the obtained gap-filled sequence and annotates it with a quality score. The main steps are illustrated in Fig. 1.

Read subsampling The first step requires an indexed BAM file of linked reads mapped on the draft assembly and an indexed Fastq file. For each gap, it extracts the linked reads whose barcode is observed in chunk regions surrounding the gap, using the thirdparty tool LRez [18]. The chunk region size can be defined by the user, the default value being 5,000 bp. To increase specificity, we keep only the barcodes for which the number of occurrences in the union set from the two flanking sequences is larger than a user-defined parameter *-f* (by default 2). The goal of this step is to get a subsample of reads that will be used in the local assembly step, instead of using the whole set of reads, thus reducing the complexity of the assembly graph and the running time.

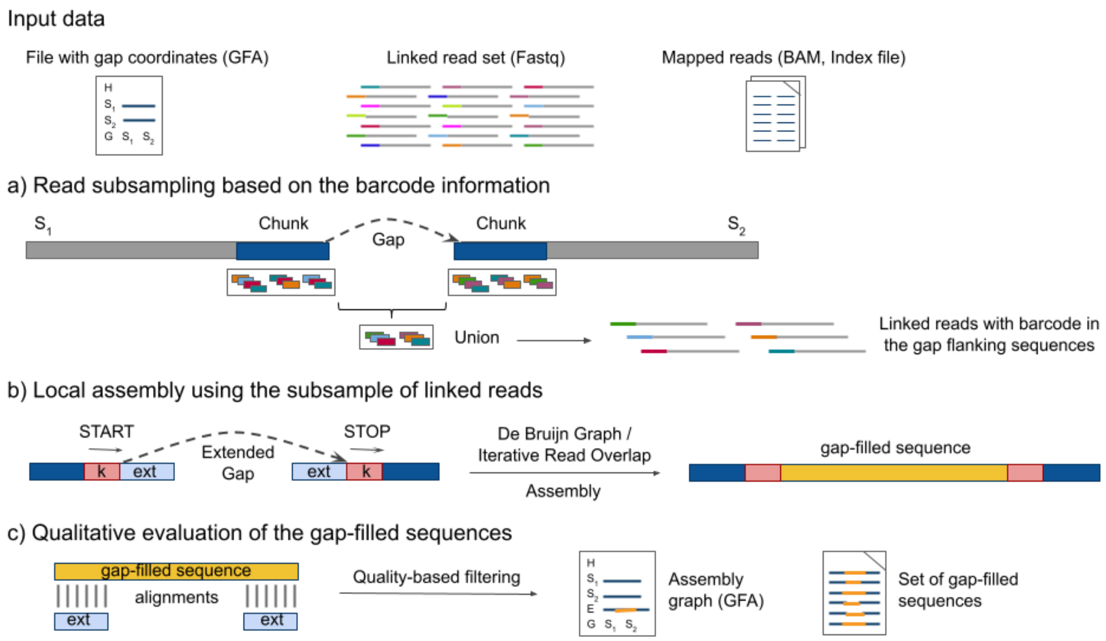
Input data
File with gap coordinates (GFA)
H
S_1
S_2
G S_1 S_2
Linked read set (Fastq)
Mapped reads (BAM, Index file)
a) Read subsampling based on the barcode information
S_1
Chunk
Gap
Chunk
S_2
Union
Linked reads with barcode in the gap flanking sequences
b) Local assembly using the subsample of linked reads
START
Extended Gap
STOP
k
ext
ext
k
De Bruijn Graph / Iterative Read Overlap
Assembly
gap-filled sequence
c) Qualitative evaluation of the gap-filled sequences
gap-filled sequence
alignments
ext
ext
Quality-based filtering
H
S_1
S_2
E
G S_1 S_2
Assembly graph (GFA)
Set of gap-filled sequences

Fig. 1. Overview of the MTG-Link gap-filling pipeline. a) Linked reads whose barcode is observed in chunk regions surrounding the gap are extracted, and constitute the read subsample used in the local assembly step. b) The local assembly is performed on an extended gap, from the k-mer *START* (source) to the k-mer *STOP* (target), using the subsample of linked reads obtained in (a). c) A quality score is assigned to the gap-filled sequence according to its alignment against the gap flanking sequences. Only the gap-filled sequences with good quality scores are returned.

Local assembly To fill the gap between two contigs, we perform a local assembly using the subsample of linked reads obtained during the first step. The goal is to find a path between the source sequence and the target sequence surrounding the gap, using an assembly algorithm. To be able to further evaluate the obtained gap-filled sequence, we extend the gap on both sides by *-ext* bp (by default 500 bp). Thus, MTG-Link will perform the local assembly between the sequences surrounding the extended gap, e.g. from the k-mer *START* (source) to the k-mer *STOP* (target). Two assembly algorithms can be used during this step: the *DBG algorithm* or the *Iterative Read Overlap algorithm.*

The *DBG algorithm* is performed with the *fill* module of the software MindTheGap [17]. MindTheGap was originally developed for the detection and assembly of insertion variants, but it also includes an efficient local assembly module (*fill* module) that relies on a De Bruijn graph data structure to represent the input read sequences. Basically, starting from a source k-mer, it performs a breadth-first traversal of the De Bruijn graph, building a contig graph. The traversal is halted when the contig graph becomes too complex. Then, all the contigs in the graph are searched for the presence of the target k-mer. If one or more contigs are found containing the target k-mer, it returns all possible sequence paths between both k-mers. In MTG-link, it is then used to perform a local assembly for each pair of gap-flanking k-mers. In MindTheGap, as in any De Bruijn graph based assembly, two parameters have major impacts on the quality of the assembly: the k-mer size and the k-mer abundance threshold for including a k-mer in the graph (solid k-mer threshold). These parameters are usually set in accordance with the expected sequencing depth. In the case of MTG-link, the latter may vary depending on the efficiency of the barcode-based subsampling step. Hence for higher sensitivity, MTG-Link automatically tests different values for these two parameters, starting with the highest ones and decreasing the values if no inserted sequence with good quality is found.

MTG-Link integrates another assembly algorithm: the *Iterative Read Overlap algorithm.* This algorithm is based on on-the-fly computations of read overlaps and iterative extensions of the current assembly sequence. Overlapping reads are reads whose prefix (or reverse complement of the suffix) aligns with the suffix of the current assembly sequence with at most *-dmax* differences (including substitutions and indels) over at least -O_{min} bp. These overlaps are found using a seed-and-extend schema, combining a seed indexing with a hash table and a banded dynamic programming semi-global alignment algorithm. At each iteration, several possible extensions may be found, due to sequencing errors and/or repeats. In this case, the algorithm groups the overlapping reads together according to their extension sequence, and gives the priority to the longest overlap. To avoid including sequencing errors, only extensions that are supported by a minimum number of reads (parameter *-a*, by default 2) are considered. Then, another extension phase begins. When no overlapping read is found, or if there is no extension shared by a sufficient number of reads, or if the maximal assembled sequence size (user defined parameter) is reached, then the algorithm backtracks and tries other extensions previously encountered but not yet explored. Finally, if during an extension phase, the k-mer *STOP* is found, the assembly sequence is returned and the exploration ends.

Qualitative evaluation Each gap-filled sequence obtained during the local assembly step is evaluated to infer its quality and provide a score that might help filtering out putative erroneous sequences. The evaluation is based on the comparison of the gap-filled sequence to the gap flanking sequences, e.g. the sequences corresponding to the extensions of the gap *-ext*. Alignments are performed with *Nucmer* [19]. Then, MTG-Link assigns a two-letters quality score to each gap-filled sequence. The first letter represents the alignment to the left flanking sequence, and the second letter represents the alignment to the right flanking sequence. To have a good quality score, the gap-filled sequence must be larger than twice *-ext bp*, and it must align on at least 90% of the lengths of the gap flanking sequences. Otherwise, the gap-filled sequence obtained is assigned a bad quality score and is considered as erroneous. Only the gap-filled sequences with a good quality score are returned.

Implementation and availability MTG-Link is written in Python 3. In order to speed up the process, it uses a trivial parallelization scheme by giving each gap to a separate thread. MTG-Link is available on GitHub (`https://github.com/anne-gcd/MTG-Link`) under the GNU Affero GPL licence, and as a Bioconda package (`https://anaconda.org/bioconda/mtglink`). Additional Python scripts for converting input and output files to the desirable formats are also provided.

2.2 Validation of the method with simulated gaps

Simulated gaps We evaluated our method with a real linked read dataset but with simulated gaps in the assembly, for which we know the true sequence to be assembled (hereafter called reference sequence) in order to assess the quality of the results. One individual genome of the butterfly *Heliconius numata* was sequenced with the 10X Genomics Chromium technology and was assembled with Supernova [8] in a draft genome assembly (genome size of ∼320 Mb) [20] (BioProject PRJNA676017, individual 37). The number of reads in the dataset is approx. 110 million, with an effective read depth of 40X. We tested MTG-Link on four different gap sizes (1, 5, 10 and 20 Kbp). For each gap size, we simulated 57 gaps in the draft assembly.

MTG-Link parameters MTG-Link was used in version 1.1.0 with the same set of parameters for all gaps. For the read subsampling step, we tested different chunk sizes (5, 10 and 15 Kbp). For the local assembly step, we used the *DBG algorithm*, with a k-mer size of [61, 51, 41, 31, 21] and a solid k-mer threshold of [3, 2]. The extension size chosen was 500 bp.

Evaluation In order to evaluate the quality of the results, we performed *Blastn* [21] alignments of each obtained gap-filled sequence to the corresponding reference sequence. The gap-filled sequences having more than 85% identity and coverage with the reference sequence are labelled as ”successful”. However, if they have less than 85% identity and coverage with the reference sequence, they are considered as ”erroneous”. The ”no gap-fillings” represent those for which no gap-filled sequence with a good quality score was found, e.g. no solution was returned by MTG-Link.

Comparison with other approaches To assess the impact of the read subsampling on the quality of the gap-filling, the running time and the memory consumption, we compared the results obtained with MTG-Link to those obtained with MindTheGap. As MTG-Link was run with the *DBG algorithm*, the local assembly step is the same in both approaches. The two approaches differ by the read subsampling and the qualitative evaluation steps which are specific to MTG-Link. Besides, as the read coverage can be highly variable in MTG-Link due to the read subsampling step, different *DBG* parameters values are automatically tested. On the contrary, as the whole set of reads is used for the local assembly in

MindTheGap, it was run with a unique parameter set: k-mer size (*-k*) of 51 and solid k-mer threshold (*-a*) of 3.

2.3 Application on real gaps of *Heliconius numata* genomes

We applied MTG-Link to the gap-filling of the Supergene P locus (1.3 Mbp) of the butterfly *Heliconius numata*. Twelve individuals genomes with different haplotypes were sequenced with the 10X Genomics Chromium technology and were assembled in draft genome assemblies with the Supernova assembler [20]. The number of reads in each dataset is approx. 110 million, with an effective coverage ranging from 20X to 47X (BioProject PRJNA676017). We attempted to fill the gaps between scaffolds of the Supergene P locus in eight individuals, for which this locus was fragmented. For this purpose, we re-scaffolded this locus by analyzing shared barcodes between scaffolds, and performed gap-filling with MTG-Link. MTG-Link was used with the *DBG algorithm*, with a k-mer size of [61, 51, 41, 31, 21] and a solid k-mer threshold of [3, 2]. For all other parameters, default values were used.

3 Results

3.1 Validation with simulated gaps

MTG-Link was assessed on simulated gaps of various sizes from a real linked read dataset of one *H. numata* genome. For each gap size (1, 5, 10 and 20 Kbp), we applied our tool on a GFA file containing 57 gaps. The results obtained with MTG-Link are represented by the right bars on each subplot in Fig. 2.

Among all tested gap sizes (228 gaps in total), 189 gaps were completely filled with MTG-Link and returned with a good quality score. Among them, 170 gaps have a correct assembled sequence (e.g. >85% identity and coverage with the reference sequence), hereafter referred as successful gap-fillings. Thus, MTG-Link has a precision of 90% with a recall of 75%. As we can observe in Fig. 2, the quality of the gap-filling depends primarily on the gap size. The gap-filling is mostly successful for small gaps (1 and 5 Kbp), but it is more difficult to close larger gaps (10 and 20 Kbp).

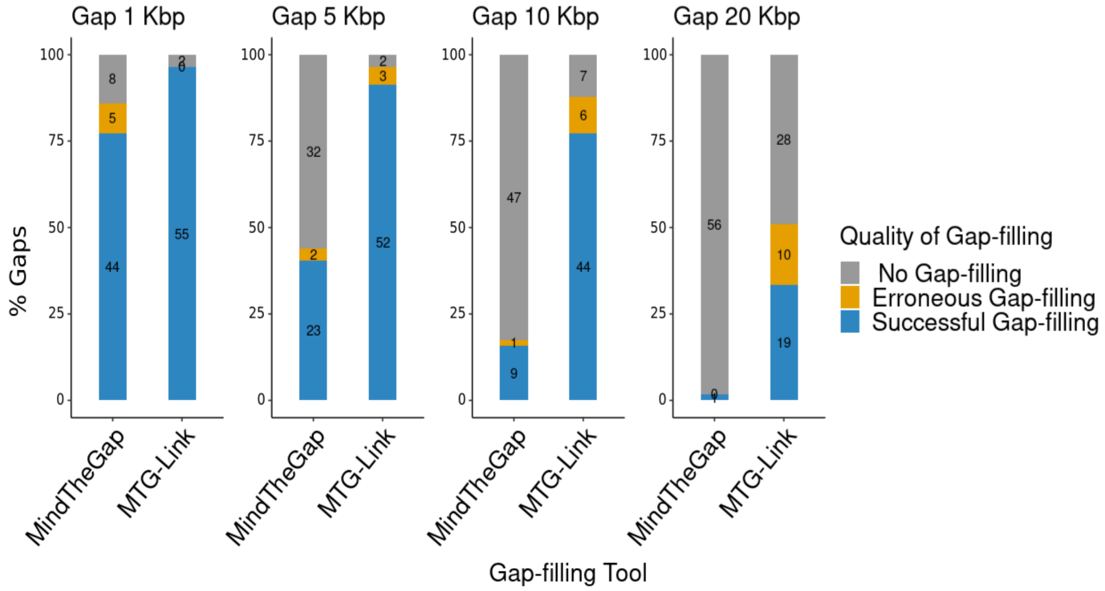

Fig. 2. Comparison of two gap-filling tools on several sets of simulated gaps. MTG-Link and MindTheGap were applied on four sets with different gap sizes, each composed of 57 simulated gaps. MTG-Link was run with the *DBG algorithm* and a chunk size of 5 Kbp.

Interestingly, we noticed that when there is no solution returned by MTG-Link (e.g. ”no gap-fillings”), in some cases the number of barcodes observed in chunk regions surrounding the gap is very small ($\leq$ 500) (Fig. 3A). However, a higher number of barcodes does not guarantee that the gap will be successfully filled. Indeed, increasing the chunk size, and consequently getting a larger number of barcodes, does not improve the gap-filling (Fig. 3B). More precisely, we observed that the gaps labelled as "no gap-fillings" but having a number of barcodes higher than 500 are those for which MTG-Link finds a solution but with a bad quality score.

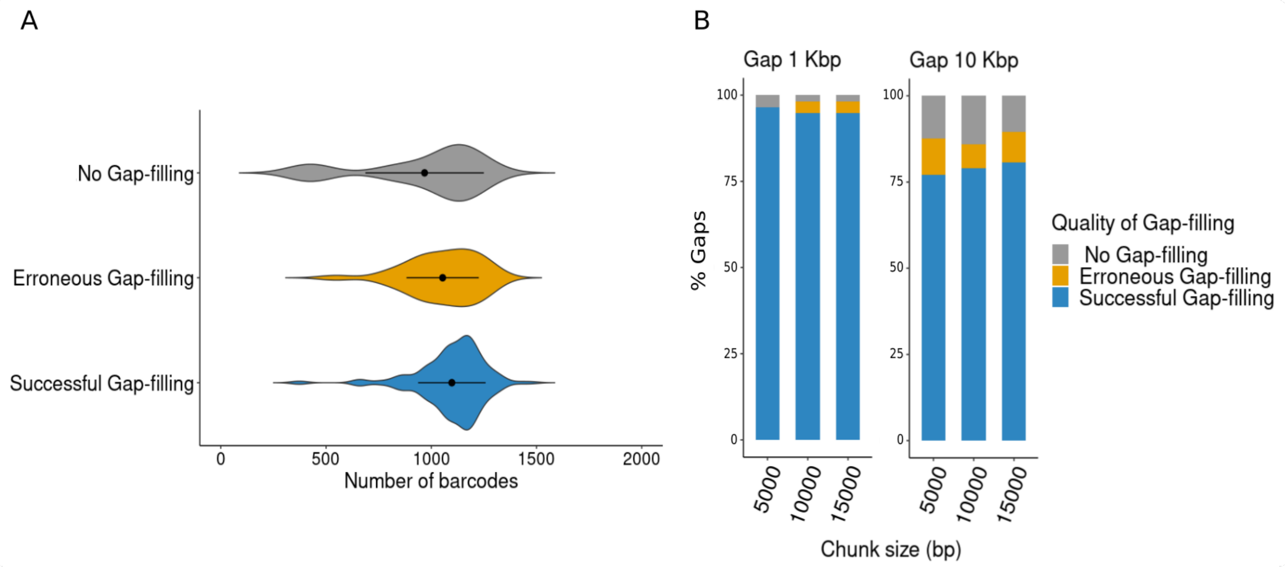

Fig. 3. Influence of two variables of the quality of the gap-filling performed by MTG-Link. A) Influence of the number of barcodes on the gap-fillings. The results shown here are obtained for all tested gap sizes (1, 5, 10 and 20 Kbp) and with a chunk size of 5 Kbp. B) Influence of the chunk size on the gap-fillings. Three different chunk sizes were tested for 1 Kbp and 10 Kbp gaps.

The erroneous gap-fillings were manually investigated. Most of the gap-fillings showed high sequence similarities with the reference sequence, but were incomplete. In several cases, we observed the presence of direct repeats in the reference sequence, generating a cycle in the De Bruijn graph whose sequence (between repeat copies) is lost in the assembly. Preliminary results obtained with the *Iterative Read Overlap algorithm* showed that this algorithm allows the correct gap-filling of some of these gaps.

The quality score assigned by MTG-Link during the qualitative evaluation step does not give a perfect auto-evaluation of the gap-filling, but it still improves its accuracy. Among the 228 tested gaps, the quality score filter enabled to discard 9 erroneous gap-fillings at the expense of losing 6 false negatives. In our method, we chose to favor precision over recall (precision of 90% with the filter vs. 86% without the filter).

Comparison with MindTheGap The gap-fillings performed by MTG-Link were compared to those obtained with MindTheGap, the tool used in the local assembly step of our pipeline. By comparing these two gap-filling tools, we are able to assess the impact of the read subsampling and the qualitative evaluation steps on the gap-filling results. Results are presented in Fig. 2. As expected, MTG-Link outperforms MindTheGap by returning more successful gap-fillings, for all tested gap sizes. Only 34% of gaps were successfully filled with MindTheGap, against 75% with MTG-Link. The differences tend to increase with the gap size. Therefore, the read subsampling and the qualitative evaluation steps greatly improve the gap-filling.

	Gap 1Kbp		Gap 5 Kbp		Gap 10 Kbp		Gap 20 Kbp	
	Time	Memory	Time	Memory	Time	Memory	Time	Memory
MTG-Link	1min27s	2.7 G	1min38s	3.1 G	2min2s	5.0 G	2min26s	13.4 G
MindTheGap	3min23s	15.1 G	3min26s	15.0 G	3min35s	15.7 G	3min34s	15.2 G

Tab. 1. Comparison of resources used by two gap-filling tools on several sets of simulated gaps. For each gap size, MTG-Link and MindTheGap were applied on a set of 57 simulated gaps. MTG-Link was run with the *DBG algorithm*. The values reported in this table are the average runtime for one gap, and the memory peak reached during each run of 57 gaps.

Importantly, MTG-Link is also significantly faster than MindTheGap. The average runtime of MTG-Link is comprised between 1.5 and 2.4 minutes per gap, which is approx. two times smaller than MindTheGap runtime (approx. 3.5 minutes per gap), as shown in Tab. 1. Although MTG-Link tests several parameters values contrary to MindTheGap, it remains faster thanks to the read subsampling step. Hence, MTG-Link is a time/memory efficient gap-filling tool.

3.2 Application on real gaps of *Heliconius numata* genomes

We applied MTG-Link on real gaps from real linked read datasets to improve the contiguity of the Supergene P locus of the butterfly *Heliconius numata*. The Supergene P locus is a locus of biological interest in *H.numata* as it controls the mimetic wing colour pattern and is subject to rearrangement polymorphism [20]. Out of the twelve individual genomes sequenced and assembled in this study, the Supergene P locus was reconstructed as a single scaffold for four individual genomes. For the other eight individual genomes, the assembly of this locus was fragmented into several scaffolds (61 gaps in total). For each of these eight individuals, we attempted to fill the gaps between the scaffolds using MTG-Link. We succeeded in reducing the number of scaffolds in the Supergene P locus for all *H. numata* individuals. For two of them, the Supergene P locus was reconstructed as a single scaffold in one step of gap-filling. For the others, the assembly was still fragmented and it required additional steps of extra contigs recruitment. Finally, after all these steps, we succeeded in filling 43 out of the 61 initial gaps with MTG-Link. This improved contiguity will allow a finer analysis of the genomic structural diversity in this locus.

4 Discussion and Conclusion

In this work, we provide a novel gap-filling tool for linked read data, called MTG-Link. This tool is composed of three main steps: read subsampling, local assembly and qualitative evaluation. To our knowledge, this is the first gap-filling tool for draft genome assemblies, dedicated to linked read data. We have therefore compared our tool MTG-Link to a generic short-read local assembly tool, MindTheGap. Both use the same *De Bruijn Graph* assembly algorithm, allowing to assess the benefit of the additional read subsampling step of MTG-Link prior to local assembly. We have shown that MTG-Link outperforms MindTheGap, in terms of both time and gap-filling quality.

Therefore, this analysis highlights the main benefit of using linked read data for the gap-filling of draft genomes, as the barcode information contained in the reads allows the enrichment of reads originating from the gap region in the read set used for the assembly. By discarding a large fraction of reads originating from other regions of the genome, we reduce the noise and complexity in the assembly graph, thus making the search for the gap-filling path easier.

A valuable feature of MTG-Link is to assign a qualitative score to each gap-filled sequence. This feature allows the pipeline to automatically test several parameters values for local assembly and to select the best solution. This is important in the context of barcode-based read subsampling, as the resulting sequencing depth and thus the optimal assembly parameters values can greatly vary between gaps. Moreover, the qualitative evaluation also allows the user to choose to prioritize the precision over the recall by using a more stringent quality score, and reciprocally.

One of the characteristics of MTG-Link is that it can use either a *De Bruijn Graph (DBG) algorithm* or an *Iterative Read Overlap algorithm* in the local assembly step. For the moment, MTG-Link was mainly tested with the *DBG algorithm*, and we have shown that this algorithm performs well, especially on small gaps. However, the gap-filling is less successful on larger gaps probably due to an increased likelihood of containing some repeated regions or a drop of sequencing depth as the distance to the gap extremities grows. In this context, the *Iterative Read Overlap algorithm* appears as a promising avenue for improvement, since it allows for variable size overlaps between reads. Longer overlaps allow to disentangle repeats larger than the k-mer size used in the de Bruijn graph but smaller than the read size, whereas smaller overlaps allow the assembly of regions of the gap covered by fewer selected reads.

Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 764840, and from the French ANR ANR-18-CE02-0019 Supergene grant. We acknowledge the GenOuest bioinformatics core facility (https://www.genouest.org) for providing the computing infrastructure. We warmly thank Mathieu Joron and Paul Jay for sharing their data and results on *H. numata* Supergene locus.

References

[1] Chaisson M.J.P., Wilson R.K., and Eichler E.E. Genetic variation and the de novo assembly of human genomes. *Nat Rev Genet.*, 16(11):627–640, 2015.

[2] Chaisson M.J.P. et al. Resolving the complexity of the human genome using single-molecule sequencing. *Nature*, 517(7536):608–611, 2015.

[3] Sedlazeck F.J. et al. Accurate detection of complex structural variations using single-molecule sequencing. *Nat Methods*, 15(6):461–468, 2018.

[4] Koren S. et al. Hybrid error correction and de novo assembly of single-molecule sequencing reads. *Nat Biotechnol.*, 30(7):693–700, 2012.

[5] Lee H. et al. Third-generation sequencing and the future of genomics. *BioRxiv*, 2016.

[6] Kuleshov V., Snyder M.P., and Batzoglou S. Genome assembly from synthetic long read clouds. *Bioinformatics*, 32(12):i216–i224, 2016.

[7] Ott A. et al. Linked read technology for assembling large complex and polyploid genomes. *BMC Genomics*, 19(651), 2018.

[8] Weisenfeld N.I., Kumar V., Shah P., Church D.M., and Jaffe D.B. Direct determination of diploid genome sequences. *Genome Res.*, 27(5):757–767, 2017.

[9] Zheng G.X.Y. et al. Haplotyping germline and cancer genomes with high-throughput linked-read sequencing. *Nat Biotechnol.*, 34(3):303–311, 2016.

[10] Spies N. et al. Genome-wide reconstruction of complex structural variants using read clouds. *Nat Methods*, 14(9):915–920, 2017.

[11] Chen Z. et al. Ultra-low input single-tube linked-read library method enables short-read second-generation sequencing systems to routinely generate highly accurate and economical long-range sequencing information. *Genome Res.*, 30(6):898–909, 2020.

[12] Meier J.I. et al. Haplotype tagging reveals parallel formation of hybrid races in two butterfly species. *BioRxiv*, 2020.

[13] Luo R. et al. SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler. *GigaScience*, 1(1):18, 2012.

[14] Paulino D. et al. Sealer: a scalable gap-closing application for finishing draft genomes. *BMC Bioinformatics*, 16(1):230, 2015.

[15] Nadalin F., Vezzi F., and Policriti A. GapFiller: a de novo assembly approach to fill the gap within paired reads. *BMC Bioinformatics*, 13(Suppl 14):S8, 2012.

[16] Chu C., Li X., and Wu Y. GAPPadder: a sensitive approach for closing gaps on draft genomes with short sequence reads. *BMC Genomics*, 20(Suppl 5):426, 2019.

[17] Rizk G., Gouin A., Chikhi R., and Lemaitre C. MindTheGap: integrated detection and assembly of short and long insertions. *Bioinformatics*, 30(24):3451–3457, 2014.

[18] Morisse P., Lemaitre C., and Legeai F. LRez: C++ API and toolkit for analyzing and managing Linked-Reads data. arXiv:2103.14419, 2021.

[19] Marçais G. et al. MUMmer4: A fast and versatile genome alignment system. *PLoS Comput Biol.*, 14(1):e1005944, 2018.

[20] Jay P., Chouteau M., Whibley A., Bastide H., Parrinello H., Llaurens V., and Joron M. Mutation load at a mimicry supergene sheds new light on the evolution of inversion polymorphisms. *Nature Genetics*, 53(3):288–293, 2021.

[21] Altschul S.F. et al. Basic local alignment search tool. *J. Mol. Biol.*, 215(3):403–410, 1990.