Reference-guided genome assembly in metagenomic samples

Cervin Guyomar, Wesley Delage, Fabrice Legeai, Christophe Mougel, Jean-Christophe Simon, Claire Lemaitre

Reference-guided genome assembly in metagenomic samples

Cervin Guyomar[1,2], Wesley Delage[1], Fabrice Legeai[1,3], Christophe Mougel[3], Jean-Christophe Simon[3] and Claire Lemaitre[1]

[1] Univ Rennes, Inria, CNRS, IRISA, F-35000 Rennes, France
[2] iDiv, Deutscher Pl. 5E, 04103 Leipzig, Germany
[3] INRA EGI, F-35000 Rennes, France

Corresponding author: cervin.guyomar@idiv.de

Abstract *Assembling genomes from metagenomic data is a challenging task, because of both the many species coexisting in the samples and the polymorphism within these species. Most approaches consist in a complete assembly of the metagenome into contigs, that can then be binned into taxonomic units. On the opposite, we present in that work a targeted assembly approach in two steps. First, taking advantage of a potentially distant reference genome, a subset of the metagenomic reads is assembled into specific contigs. Then, using an enhanced version of the MindTheGap local assembly algorithm, this first draft assembly is completed using the whole metagenomic readset in a de novo manner. The resulting assembly can be output as a genome graph, allowing to distinguish different strains with potential structural variants coexisting in the sample. MindTheGap was applied to 32 pea aphid re-sequencing samples in order to recover the genome sequence of its obligatory bacterial symbiont, Buchnera aphidicola. It was able to return high quality assemblies (one contig assembly in 90% of the samples), even when using increasingly distant reference genomes, and to retrieve large structural variations in the samples. Due to its targeted approach, it outperformed standard metagenomic assemblers in terms of both time and assembly quality. As such, it appears as a promising approach for single genome assembly from metagenomic data.*

***License**: GNU Affero general public license*

***Availability:** https://github.com/GATB/MindTheGap*

Keywords Genome assembly, metagenomics, reference-guided, short reads

1 Introduction

The advances of molecular techniques revealed the importance of microorganisms in every ecosystem. In particular, whole-genome metagenomic sequencing makes it possible to understand the full functional potential of microbial communities by accessing the whole genomic sequence of both culturable and unculturable microbes. However, extracting relevant information from complex metagenomic datasets is a challenging task. Current metagenomic datasets are a mixture of short reads originating from different species. Thus, reconstructing genomes from metagenomic data requires two steps : the assembly of reads into longer sequences, and the partitioning of sequences based on their taxonomic origin.

Metagenomic assembly consists in forming contigs prior to the taxonomic binning of sequences. Many recent software are devoted to this task [1,2,3]. However, because of the high complexity of such data, *de novo* assembling contigs from metagenomic reads is challenging and comes with a high computational cost. Metagenomic assemblies are very fragmented because of homologous regions between microbial species and polymorphism within the species [4].

An alternative to this approach would be to partition in a first step the metagenomic reads into subsets assigned to different species. Binning methods relying on the nucleotidic composition of reads cannot be applied to the current Illumina reads because of their short length [5]. Alternatively, it is possible to select reads by reference-based approaches, but these approaches struggle to classify reads from badly known species, and hardly scale up to large datasets when based on alignment methods [6]. A relevant strategy to assemble a given genome from metagenomic data is to map reads against the closest available reference genome to assemble new contigs. The quality of the assembly is therefore highly dependent on the evolutionary distance with the reference genome. In particular, any region absent or too divergent from the reference genome will be missed. This enables nonetheless

the targetted assembly of a genome of interest within a community, which is for instance relevant for the study of key players of host-symbiont relationships or the discovery of new pathogenic strains of known microbes. In these use cases, functional, structural or phylogenetic genomic analyses require the assembly of a new genome of interest from metagenomic data. In that context, neither *de novo* metagenomic assembly nor assembly from reads selected by reference alignment are able to return assemblies of good quality. Nonetheless, it seems possible to use the best of these two strategies, by selecting reads from regions of homology with a related reference genome, and using *de novo* assembly to reconstruct the missing regions.

Several tools, such as MITObim [7], LOCAS [8], Pilon [9] or IMR/DENOM [10], were designed following this idea, combining reference alignment and *de novo* assembly. However, all of them show some limitations because of which they are not adapted to metagenomic data. They are either not scaling up with these large datasets (MITObim, LOCAS), unable to deal with large structural variants (IMR/DENOM, Pilon, LOCAS) or to return coexisting variants (LOCAS, IMR/DENOM).

In this work, we present a solution for the assembly of a genome of interest from metagenomic data, in a reference-based manner. This method can recover large regions absent from the given reference genome, makes no assumption on the ordering or direction of regions homologous with the reference and is able to return several different solutions reflecting the metagenomic diversity inside the sample. The method is based on two main steps, a reference based recruiting and assembly of metagenomic reads, followed by a targetted assembly, filling the gaps between the contigs assembled beforehand.

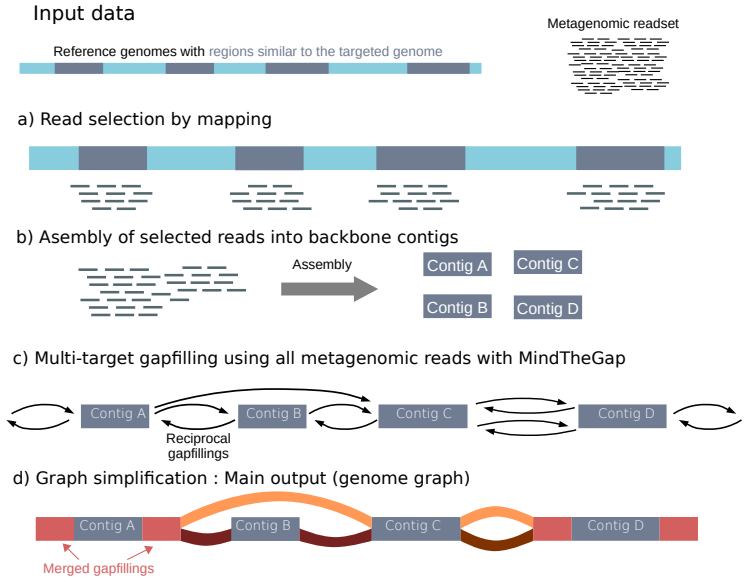
We applied this method to reconstruct genomes from several metagenomic samples of the pea aphid *Acyrthosiphon pisum*. Focusing on the primary endosymbiont *Buchnera aphidicola*, we demonstrated the ability of *MindTheGap* to assemble complete bacterial genomes in a single contig using a remote genome as a primer, even when structural variability is present.

2 Material and Methods

2.1 Targeted assembly for metagenomic data

Strategy overview The method described in this work relies on a two-step pipeline, described in Figure 1.

Fig. 1. Overview of the *MindTheGap* reference-guided assembly pipeline



The first step uses a given reference genome to build an incomplete but trustworthy assembly, matching with the conserved regions of the genome. The second step uses the whole set of metagenomic reads to extend the previously assembled contigs and form a complete assembly, without any *a priori* on the order and orientation of contigs. The result of the pipeline is a genome graph encompassing the stuctural diversity detected on the assembled genome. This graph can be exploited by extracting contigs, or paths of the graph that represent different strains.

2.1.1 Assembly of backbone contigs The first step requires a metagenomic readset and a reference genome, and returns contigs that are assembled using reads mapped on the reference. All metagenomic reads are mapped against the reference genome using BWA MEM [11], and the mapped reads are kept and *de novo* assembled using the Minia [12] assembler. Although any assembler can be used in this step, we use Minia [12] for its low memory footprint, and its assembly algorithm similar to the one used in the second step of the method. The goal of this step is to generate high quality contigs, that can reliably be used for the upcoming gapfilling. To ensure this, we set up Minia with more stringent parameters than for an usual assembly task and only contigs longer than a user-defined threshold (500 bp by default) are kept.

2.1.2 Parallel gapfilling with MindTheGap The essential step of the pipeline is the gapfilling between backbone contigs, which enables the assembly of regions absent from the reference genome. This is made possible by a targeted assembly of the whole readset using the previously assembled contigs as primers. This step does not require the ordering of contigs, since all possible combinations are tested during gapfilling. As a result, structural variants can be detected, either compared to the reference genome or within the sample.

This step is based on a module of the software *MindTheGap*, originally developed for the detection and assembly of insertion events [13]. The *fill* module of *MindTheGap* performs a local assembly for each pair of breakpoint event kmers, resulting in one or several insertion sequences.

In this work, we took advantage of this module of *MindTheGap* and adapted it to the problem of reference-guided assembly. It has been modified to make possible the gapfilling between a seed kmer and **multiple** target kmers, enabling the "all versus all" gapfilling within a set of contigs with only a linear increase of the runtime (compared to a quadratic increase for a naive "all versus all" gapfilling). The resulting algorithm is presented in Figure 2. A seed kmer is extracted at the end of each contig and its reverse-complement, resulting in a set of $2n$ kmers for n contigs. Similarly, a set of $2n$ target kmers is extracted at the beginning of each contig and its reverse-complement. For each seed kmer, a contig graph is created by starting from the seed kmer and performing a breadth first traversal of the *De Bruijn* graph representation of the whole readset. Contigs are consensus sequences returned by removing graph motifs such as bubbles (SNPs) and tip-ends (errors). In the contig graph, contigs are nodes, and edges represent the existence of a $k-1$ nucleotide overlap between two contigs. The creation of the contig graph is similar to the one used in *Minia* [12]. The traversal is stopped when the graph becomes too large (total assembled nucleotides) or too complex (number of contigs), following user-defined parameters. Importantly, if one of the target kmers is found during the contig graph construction, that contig is not extended further, avoiding redundant contig assembly, and saving time and memory. After the contig graph has been built, target kmers are searched within this contig graph, and gapfilling sequences are built, by retraversing the contig graph from the seed kmer to contigs containing a given target kmer. For each seed-target couple, if several solutions are returned, redundant solutions above a 95% identity threshold are removed. Thanks to this multi-target version of the algorithm, only $2n$ contig graph constructions are necessary to search possible sequences between all pairs of contigs, instead of n^2 with the naive approach.

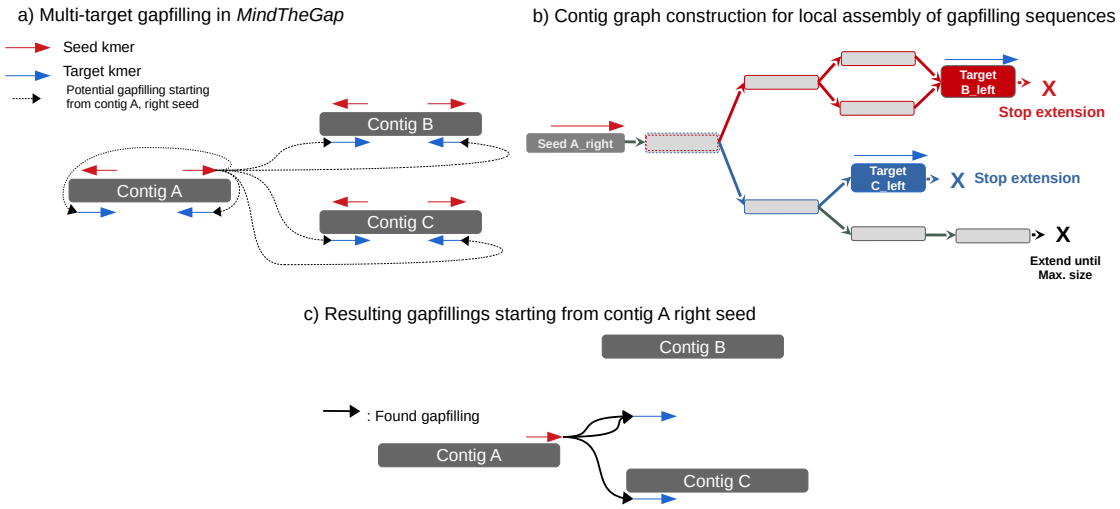
The whole process is parallelized by dispatching the $2n$ starting kmers to different threads. The main output is a genome graph in the GFA format (Graphical Fragment Assembly, https://github.com/GFA-spec/GFA-spec), giving the overlap relationships between contigs an their gapfillings.

2.1.3 Graph simplification and visualisation In order to return a standard fasta assembly, the genome assembly has to be processed. The complexity of the graph is reduced on several steps using a post-treatment program.

First, it is likely that two contigs are linked in the graph by two gapfillings with reverse-complement sequences, one starting from the left contig and the other one starting from the right contig. Such reciprocal links are removed, when their sequence identity is over a 95% threshold.

Secondly, when several gapfilling sequences start (or end) from the same seed, it is possible that a subset of them have an identical prefix (suffix) and start to diverge after a potential large distance to the seed. This results in redundant sequences in the graph. A node merging algorithm is applied, in order to return contigs that do not share large identical subsequences (prefix or suffix). Sets of

Fig. 2. Gapfilling a set of contigs using *MindTheGap* fill module.
a) Seed and target kmers are extracted from the 3 input contigs, resulting in 2 sets of 6 kmers, seed (red) and target (blue) ones. b) A graph of contigs is built starting from the right seed kmer of contig A. Extension is stopped when a target kmer of another contig if encountered, or a maximum assembly size is reached. c) This results in 3 gapfilling sequences starting from contig A right seed, 2 gapfilling sequences joining contig B, and one contig C.



sequences sharing the same 100 first nucleotides are built. Within each set, the sequences are then compared to find the first divergence between all sequences. A new node is added to the graph, containing the repeated portion of the sequences, and repeated nodes are shortened accordingly. This process is applied iteratively to every node, including the newly created nodes, for which a subset of neighbors may still show identical sequences.

Finally, simple linear paths in the graph are merged, nodes whose length is lower than 500 bp are removed, and highly branching nodes (connected to more than 5 contigs) are cut. The resulting graph is a good representation of the *MindTheGap* assembly, and nodes can be extracted to be used as regular contigs.

After the simplification process, the graph may not be a linear sequence because of intra-sample polymorphism or assembly uncertainties. The final assembly can be generated either by manual inspection of the graph using the *Bandage* software [14], or by enumerating all possible paths within the graph.

2.1.4 Implementation and availability *MindTheGap* has been officially released in version 2.1, enabling the so-called "contig mode" for reference-guided assembly (https://github.com/GATB/MindTheGap). *MindTheGap* is written in C++ using the GATB library [15] (https://github.com/GATB/gatb-core). The GATB library provides algorithms for the analysis of NGS datasets with high performances and a low memory footprint. The graph simplification is performed using Python scripts, available on the *MindTheGap* repository. A complete pipeline including mapping, assembly and gapfilling is also available as a Python script distributed along with *MindTheGap*.

2.2 Application to pea aphid metagenomic datasets

In this study, we applied *MindTheGap* assembly pipeline to the assembly of the obligatory bacterial symbiont of the pea aphid holobiont, *Buchnera aphidicola*. We considered 32 pea aphid resequencing samples of paired end 100bp *Illumina* reads. These datasets have already been studied in a previous work, in which the microbiota of each aphid sample was detailed [16]. The number of reads per dataset is ranging from 65 to 118 million, with an average coverage of 628X for the *Buchnera* genome.

Reference genomes with increasing levels of divergence Reference-guided assembly was performed with 4 distinct reference genomes of *Buchnera aphidicola* with different levels of divergence : 1) *Buchnera aphidicola* from A pisum (LSR1 accession), hereafter called *Buchnera LSR1*, which is the closest available assembled genome; 2) *Buchnera* from *Myzus persicae*; 3) *Buchnera* from *Uroleucon*

ambrosiae the most divergent reference analyzed ; and 4) a synthetic genome obtained by deleting 116.4 Kb of sequences from *Buchnera LSR1*. The synthetic rearranged LSR1 genome was generated by applying 20 deletions, whose size ranged from 300 bp. to 20 kbp. The levels of divergence are supported by phylogenetic studies [17] and genome alignment. *Buchnera LSR1* was aligned on the *Myzus persicae* with a 93% coverage, and to *Uroleucon ambrosiae* witha 87% coverage, with a genome identity of 80% on the aligned regions.

Inclusion of simulated structural variations To assess the ability of *MindTheGap* to recover structural variations in samples with strain diversity, we created a synthetic pea aphid sample by adding to a randomly chosen real sample, a subset of simulated reads from the previously described rearranged genome (with 20 deletions). 50X coverage of reads were simulated with *wgsim* of the Samtools suite.

MindTheGap assembly pipeline parameters *MindTheGap* was used in version 2.2.0, with the same set of parameters for all samples and reference genomes. For the assembly step, a *kmer* size of 61 was chosen, along with a solidity threshold of 10, and a minimum contig length of 400 bp. The gapfilling step was performed using a k value of 51, and a solidity threshold of 5.

Comparison with other approaches The results were compared to those of a usual approach to assemble a particular genome from metagenomic data. A complete *de novo* assembly was performed for each sample using MegaHit [3] and *Buchnera* contigs were selected by a Blast alignment against the genome of *Buchnera aphidicola APS*. Only contigs with at least 50% of the length covered by Blast hits with e-value smaller than 10^{-5} were kept.

The quality of each assembly was assessed using Quast [18] and the reference genome of *Buchnera aphidicola APS* from *A. pisum*. Similarly to what was done with *MindTheGap*, we did not include contigs smaller than 1 Kb, mainly associated with plasmid sequences.

3 Results

3.1 Single chromosome assembly of *Buchnera aphidicola* from metagenomic data

MindTheGap assembly pipeline was applied on 32 pea aphid resequencing samples [16] to assemble its bacterial obligatory symbiont *Buchnera aphidicola* (640 Kb). These are metagenomic samples comprising the insect host genome together with is microbial symbiotic communities. More than 90% of the reads originate from the insect host, and are not relevant when focusing on symbiont genomes. This particular fact motivates the choice of a targeted assembly technique, which does not require to assemble all the pea aphid reads.

In order to assess the robustness of the approach with respect to the level of divergence of the reference genome, four different genomes of *Buchnera aphidicola* of increasing divergence were used as a guide for the assembly, and the resulting contigs were compared to the closest reference available as a validation.

A summary of the assemblies obtained using the different reference genomes is shown in Table 1. When using either *A. pisum (LSR1)* or *M. persicae* reference genomes, most samples were assembled in a single contig whose length is very close to the target length (Less than 1% length divergence, or 6 kb). 91% of samples were assembled in a single complete contig. Using *Buchnera* from *Uroleucon* as a guide returns less complete assemblies, with only 65% of the samples that were fully assembled. This is due to its greater evolutionary distance to the genome to assemble, This greater distance is particularly well exemplified when looking at the relative contributions between the two steps of the pipeline, mapping based assembly and *de novo* gapfilling. Only an average of 6.92% of the target genome is assembled after the first step when using *Uroleucon's Buchnera*, whereas this fraction is of 47.6 % for *Myzus* and 99.9% for *Acyrthosiphon*.

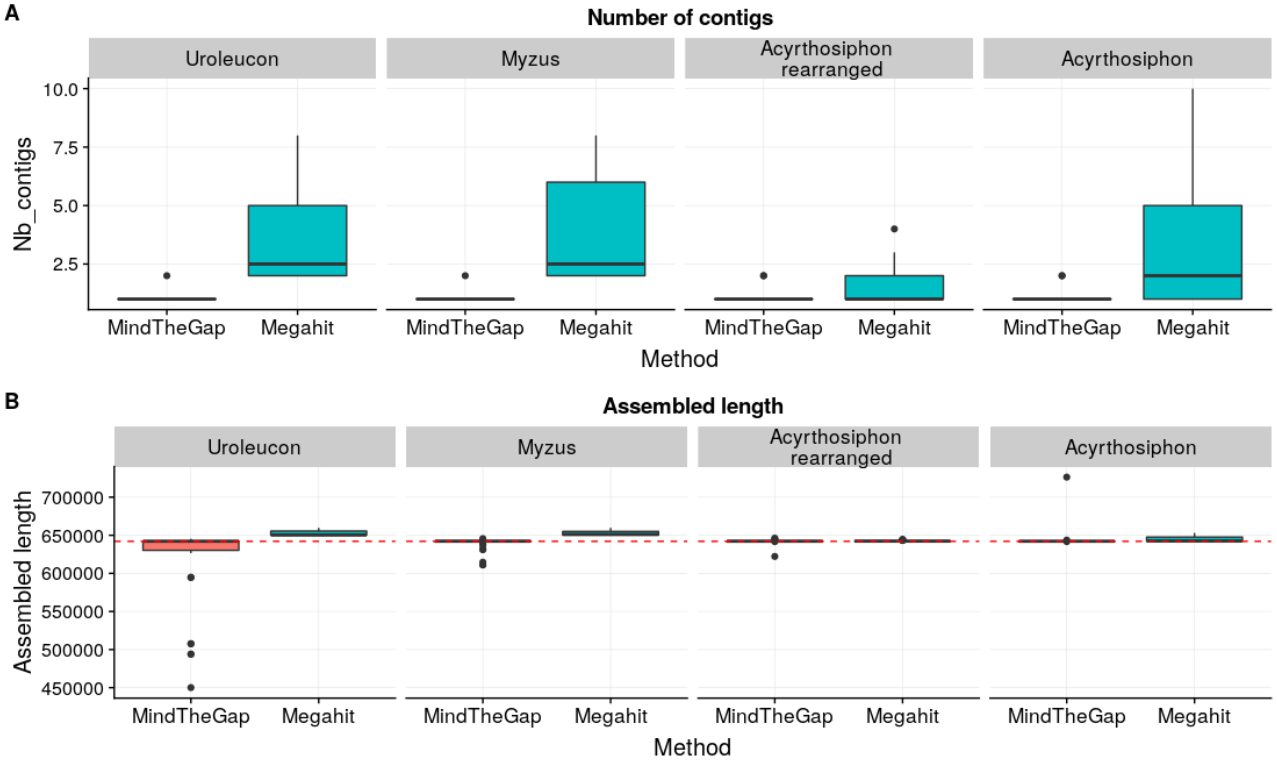
When using a rearranged genome missing several large sequences (totaling 116.4 Kb), most samples were also assembled into a single contig and all the missing regions were fully recovered. Although the complete genome length was recovered, less circular contigs were returned compared to other reference genomes.

Comparison with a classical metagenomic assembly The assemblies performed by *MindTheGap* were compared to those of an alternative stratergy, consisting in a *de novo* assembly using *MegaHit* [3] followed by a selection of contigs using a reference genome.

	Buchnera *Uroleucon ambrosiae*	Buchnera *Myzus persicae*	Buchnera rearranged	Buchnera *Acyrthosiphon pisum*
Circular complete assemblies	20	22	17	22
Linear complete assemblies	1	5	12	7
2 contig complete assemblies	1	2	2	2
Incomplete / erroneous assemblies	9	3	1	1

Tab. 1. Overview of assembly results with four different *Buchnera* reference genomes used as guide, from the closest relative at the right, to the most distant at the left. A complete assembly has a size with no more than 1% variation compared to the reference genome *Buchnera* LSR1.

Fig. 3. Number of contigs (A) and assembly length (B) using four different *Buchnera* genomes as assembly guide. The expected genome length (*Buchnera* LSR1) is shown as a red dotted line.



For most samples, *MindTheGap* outperforms the metagenomic assembly by returning assemblies with less contigs, and a total length closer to the expected genome size. Reference-guided assembly enables a one-contig assembly in most cases (90%), whereas *MegaHit* outputs a single contig for only 28% of samples. The average assembly size for *MegaHit* exceeds the expected genome length. An explanation for this could be that highly polymorphic regions may be assembled into distinct contigs by the metagenomic assembler, while *MindTheGap* merges them, or represents them as bubbles in the genome graph.

Importantly, *MindTheGap* is also significantly faster than *MegaHit*. The average runtime of *MindTheGap* assembly pipeline is 95 minutes, which is 5.5 times inferior to *MegaHit* runtime (525 minutes). Indeed, *MegaHit* produces contigs not only for the target organism, but in this case for the insect host *A. pisum* and its secondary symbionts.

3.2 Assembly of large structural variations in a metagenomic context

MindTheGap was applied to a pea aphid sample in which simulated reads from a rearranged *Buchnera* genome were added, simulating the coexistence in a metagenomic dataset of two strains with structural variations. In the resulting genome graph, 17 out of the 20 simulated deletions were fully recovered, with both the deleted and complete versions of the genome assembled. Extracting the longest path from the graph resulted in a one contig 641,531 bp assembly, compared to the 642,011 bp of the *Buchnera* LSR1 genome. Similarly, the shortest path extracted from the graph was 526,448bp long, compared to 525,611 for the deleted simulated genome. The longest structural variations (up to 20 Kb) were all successfully recovered. Only two 500 bp and one 300 bp variations were missing from the graph.

The metagenomic assembly with *MegaHit* of the same readset, followed by a filtering of contigs using the deleted reference genome, resulted in a 38 contigs assembly, with a length of 645,973 bp and a N50 of 44,484 bp. It highlights the difficulty of *de novo* assembly to deal with structural diversity in metagenomic samples.

4 Discussion and conclusion

Starting from the observation that both reference-based assignation and *de novo* assembly are inadequate to study some aspects of the metagenomic diversity, we present in the present work an hybrid method under the term of reference-guided assembly. This method was designed to assemble the genome of a single species of interest and its structural variants from a potentially larege and complex metagenomic dataset. We have shown here that it outperforms both reference-based approaches and *de novo* assemblers. Reference based read assignation is highly dependent on the evolutionary distance of the targeted genome with available references. This was particularly highlighted in this work, where less than 10 % of the genome could be assembled with the reads mapping to the most divergent reference genome used in this analysis. In *de novo* approaches, the assembly is performed prior to contig binning or mapping. This can be described as an *Assembly-first* approach. Here, we present a *Mapping-first* approach, that lightens the computational burden of full *de novo* metagenomic assembly, at the cost of a single genome assembly. To our knowledge, this is the first reference-based assembly approach suitable for metagenomic data.

Beyond the pea aphid complex, *MindTheGap* may also be applied to a wide range of assembly issues. The targeted assembly approach reduces the number of sequences to assemble, and thus simplifies the assembly problem. This approach may therefore be suitable for large and complex communities such as the human microbiome. Here, *MindTheGap* was presented as a complete pipeline from reads to contigs, but the second step of the pipeline can be associated to any other assemblers. In this manner, *MindTheGap* can be used as a finishing tool for previous incomplete assemblies. In a metagenomic context, the gapfilling step may be a way to increase the contiguity of assemblies by joining metagenomic contigs identified by binning methods as coming from the same species.

A valuable feature of *MindTheGap* is to output a genome graph representation instead of a set of unconnected contigs. This is particularly useful to represent the structural diversity of the genomes, which is rarely examined in metagenomic datasets.

Acknowledgements

Computations have been made possible thanks to the resources of the Genouest infrastructure.

References

[1] Sergey Nurk, Dmitry Meleshko, Anton Korobeynikov, and Pavel A. Pevzner. metaSPAdes: a new versatile metagenomic assembler. *Genome Research*, 27(5):824–834, mar 2017.

[2] Yu Peng, Henry C M Leung, S. M. Yiu, and Francis Y L Chin. IDBA-UD: A de novo assembler for single-cell and metagenomic sequencing data with highly uneven depth. *Bioinformatics*, 28(11):1420–1428, jun 2012.

[3] Dinghua Li, Chi-Man Liu, Ruibang Luo, Kunihiko Sadakane, and Tak-Wah Lam. MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. *Bioinformatics (Oxford, England)*, 31(10):1674–6, may 2015.

[4] Alexander et al Sczyrba. Critical Assessment of Metagenome Interpretation - A benchmark of metagenomics software. *Nature Methods*, 14(11):1063–1071, 2017.

[5] Hanno Teeling, Anke Meyerdierks, Margarete Bauer, Rudolf Amann, and Frank Oliver Glöckner. Application of tetranucleotide frequencies for the assignment of genomic fragments. *Environmental Microbiology*, 6(9):938–947, sep 2004.

[6] Daniel H. Huson, Sina Beier, Isabell Flade, Anna Górska, Mohamed El-Hadidi, Suparna Mitra, Hans Joachim Ruscheweyh, and Rewati Tappu. MEGAN Community Edition - Interactive Exploration and Analysis of Large-Scale Microbiome Sequencing Data. *PLoS Computational Biology*, 12(6):e1004957, 2016.

[7] Christoph Hahn, Lutz Bachmann, and Bastien Chevreux. Reconstructing mitochondrial genomes directly from genomic next-generation sequencing reads - A baiting and iterative mapping approach. *Nucleic Acids Research*, 41(13):e129, jul 2013.

[8] Juliane D. Klein, Stephan Ossowski, Korbinian Schneeberger, Detlef Weigel, and Daniel H. Huson. LOCAS – A Low Coverage Assembly Tool for Resequencing Projects. *PLoS ONE*, 6(8):e23455, aug 2011.

[9] Bruce J. Walker, Thomas Abeel, Terrance Shea, Margaret Priest, Amr Abouelliel, Sharadha Sakthikumar, Christina A. Cuomo, Qiandong Zeng, Jennifer Wortman, Sarah K. Young, and Ashlee M. Earl. Pilon: An integrated tool for comprehensive microbial variant detection and genome assembly improvement. *PLoS ONE*, 9(11):e112963, nov 2014.

[10] Xiangchao Gan, Oliver Stegle, Jonas Behr, Joshua G. Steffen, Philipp Drewe, Katie L. Hildebrand, Rune Lyngsoe, Sebastian J. Schultheiss, Edward J. Osborne, Vipin T. Sreedharan, André Kahles, Regina Bohnert, Géraldine Jean, Paul Derwent, Paul Kersey, Eric J. Belfield, Nicholas P. Harberd, Eric Kemen, Christopher Toomajian, Paula X. Kover, Richard M. Clark, Gunnar Rätsch, and Richard Mott. Multiple reference genomes and transcriptomes for Arabidopsis thaliana. *Nature*, 477(7365):419–423, sep 2011.

[11] Heng Li and Richard Durbin. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, 25(14):1754–1760, jul 2009.

[12] Rayan Chikhi and Guillaume Rizk. Space-efficient and exact de Bruijn graph representation based on a bloom filter. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7534 LNBI(1):236–248, sep 2012.

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

[14] Ryan R. Wick, Mark B. Schultz, Justin Zobel, and Kathryn E. Holt. Bandage: Interactive visualization of de novo genome assemblies. *Bioinformatics*, 31(20):3350–3352, oct 2015.

[15] Erwan Drezen, Guillaume Rizk, Rayan Chikhi, Charles Deltel, Claire Lemaitre, Pierre Peterlongo, and Dominique Lavenier. GATB: Genome Assembly & Analysis Tool Box. *Bioinformatics (Oxford, England)*, 30(20):2959–2961, oct 2014.

[16] Cervin Guyomar, Fabrice Legeai, Emmanuelle Jousselin, Christophe Mougel, Claire Lemaitre, and Jean-Christophe Simon. Multi-scale characterization of symbiont diversity in the pea aphid complex through metagenomic approaches. *Microbiome*, 6(1):181, 2018.

[17] Eva Nováková, Václav Hypša, Joanne Klein, Robert G Foottit, Carol D von Dohlen, and Nancy A Moran. Reconstructing the phylogeny of aphids (hemiptera: Aphididae) using dna of the obligate symbiont buchnera aphidicola. *Molecular Phylogenetics and Evolution*, 68(1):42–54, 2013.

[18] Alexey Gurevich, Vladislav Saveliev, Nikolay Vyahhi, and Glenn Tesler. QUAST: Quality assessment tool for genome assemblies. *Bioinformatics*, 29(8):1072–1075, apr 2013.