







Peer Community In Mathematical & Computational Biology

RESEARCH ARTICLE

 Open Access
 Open Peer-Review
 Open Data
 Open Code

HairSplitter: haplotype assembly from long, noisy reads

Roland Faure^{1,2}, Dominique Lavenier¹ & Jean-François Flot^{2,3}

Cite as:

xxx

¹ Univ. Rennes, INRIA RBA, CNRS UMR 6074, Rennes, France

² Service Evolution Biologique et Ecologie, Université libre de Bruxelles (ULB), Brussels, Belgium

³ Interuniversity Institute of Bioinformatics in Brussels – (IB)², Brussels, Belgium

Correspondence:

roland.faure@irisa.fr

Recommender:

FirstName FamilyName

Reviewers:

FirstName FamilyName and
two anonymous reviewers

This version of the article has not yet been peer-reviewed by
Peer Community In Mathematical and Computational Biology
(<https://doi.org/xxx/xxx>)

Abstract

Motivation: Long-read assemblers face challenges in discerning closely related viral or bacterial strains, often collapsing similar strains in a single sequence. This limitation has been hampering metagenome analysis, where diverse strains may harbor crucial functional distinctions.

Results: We introduce a novel software, HairSplitter, designed to retrieve strains from a strain-oblivious assembly and long reads. The method uses a custom variant calling process to operate with erroneous long reads and introduces a new read binning algorithm to recover an a priori unknown number of strains. On noisy long reads, HairSplitter can recover more strains while being faster than state-of-the-art tools, both in the viral and the bacterial case.

Availability: HairSplitter is freely available on GitHub at github.com/RolandFaure/HairSplitter.

Contact: roland.faure@irisa.fr

Keywords: Metagenomes; Metaviromes; Haplotyping; Genome assembly; Strain separation

Introduction

Microbiomes play a crucial roles in many ecosystems, such as soils or human guts, in turn impacting human health (Conlon and Bird, 2014) and soil fertility (Coban et al., 2022). Microbiomes typically contain sets of organisms with highly similar genomes, the sequences of which are called haplotypes (short for “haploid genotypes” (Ceppellini et al., 1967)). Distinguishing these lineages is an important challenge, as small genomic differences between haplotypes can lead to significant phenotypic changes. For instance, some strains of *Escherichia coli* can be pathogenic or commensal while having an Average Nucleotide Identity (ANI) (Konstantinidis and Tiedje, 2005) of more than 98.5% (Frank et al., 2011). A few mutations also became famous for altering significantly the infectiousness of some coronaviruses lineages (Magazine et al., 2022).

De novo sequencing and assembling is a central method to characterize microbial communities. Unlike previous methods, it allows to analyse the composition of a metagenome without culturing the strains, enabling a wide range of analyses (Ward, 2006). While existing genome assemblers proficiently reconstruct genomes of abundant species, they struggle to distinguish viral or bacterial haplotypes. The main difficulty for assemblers lies in the unknown number of haplotypes in a sample and their uneven coverage (Ghurye et al., 2016).

Many tools have been developed to overcome this problem in the context of short-read assemblies, such as OPERA-MS (Bertrand et al., 2019), Constrains (C Luo et al., 2015), STRONG (Quince et al., 2020), StrainXpress (Kang et al., 2022) and VStrains (R Luo and Lin, 2023). However, these methods are not designed for long-read sequencing and do not exploit the long-range information contained in long reads.

Long reads with extremely low error rate, such as PacBio HiFi reads, have been used to distinguish finely strains with the help of specialized software such as hifiasm (Cheng et al., 2021) and stRainy (Kazantseva et al., 2023). However, this challenge has not been yet successfully tackled in the case of noisier reads such as “regular” PacBio data or Oxford Nanopore Technology (ONT) reads, the latter of which can be obtained very rapidly on cheap sequencers that are small enough to be carried into the field (Cesare et al., 2024; Runtuwene et al., 2019).

Several methods have been implemented to deal with haplotype separation for long reads with high error rates. While the viral and bacterial haplotype assembly problems are identical in their formulation, the characteristics of the input data vary significantly: the genomes are generally much shorter and much more deeply sequenced in the viral case. This has led to the emergence of software specialized in either of the two problems. In the context of bacterial strain separation, Vicedomini et al., 2021 showed that mainstream assemblers such as metaFlye (Kolmogorov et al., 2020) and Canu (Koren et al., 2017) failed to distinguish close bacterial haplotypes and proposed a new tool, called Strainberry, to reconstruct strains. In the context of viral strain separation, Strainline (X Luo et al., 2022) and HaploDMF (Cai et al., 2022) were presented to tackle specifically the viral haplotype reconstruction problem and need very high depth of sequencing to work. The method iGDA (Z Feng et al., 2021) was proposed as a general approach to phase minor variants while handling high error rates and can theoretically assemble both bacterial and viral haplotypes. The main shortcomings of all of these methods is that they struggle to recover haplotypes of low abundance. Additionally, most of these tools are very computationally intensive.

We present HairSplitter, an efficient pipeline for separating haplotypes in the viral and bacterial context using potentially error-prone long reads. HairSplitter first calls variants using a custom process to distinguish actual variants from alignment or sequencing artefacts, clusters the reads into an unspecified number of haplotypes, creates the new separated contigs and finally untangles the assembly graph. HairSplitter can be used for either metaviromes or bacterial metagenomes.

48

49 Methods

49

50 Overview of the pipeline

51 HairSplitter takes as input an assembly (in fasta format) or an assembly graph (in gfa format) as well as se-
52 quencing reads (fasta/q) and produces a new assembly (fasta and gfa). The HairSplitter pipeline is depicted on
53 Figure 1 and comprises five steps: 1) correcting the assembly, 2) calling variants on each contig, 3) separating
54 the reads by haplotype on each contig, 4) reassembling the strain-specific contigs and 5) unzipping.

55 Completion of the assembly graph

56 To work well, HairSplitter needs as input an assembly graph on which all non-chimeric reads align from end
57 to end, which we define as a “complete” assembly graph. If the assembly was not provided as a graph, it is
58 turned into an incomplete graph with no edges. Collapsed assembly graphs are also often incomplete because
59 of contigs that have been detached from their neighbors and of collapsed structural variation between strains.

60 Aligning reads on an incomplete graph translates as locations where a significant number of reads stop
61 aligning, which we call breakpoints. Breakpoints can occur in the middle or the end of contigs. To complete
62 the initial assembly graph, the reads are aligned on the graph using minigraph (Li et al., 2020). The assembly is
63 subsequently examined for breakpoints and HairSplitter breaks the contigs at these breakpoints. Additionally,
64 links are added in the graph between ends of contigs when there is sufficient read support. The process is
65 illustrated in Figure 1a. An evaluation of this step in terms of misassemblies and contiguity is provided in
66 Supplementary Table 5.

67 The completed assembly resulting from this process is used throughout the subsequent stages of the
68 pipeline.

69 Mathematical model behind variant calling

70 To sort reads into haplotypes, the intuitive method of clustering reads based on the similarity of their full
71 sequence proves ineffective due to the dominance of sequencing and alignment errors, obscuring strain differ-
72 ences. HairSplitter first identifies variant positions, pinpointing loci where strains exhibit actual differences.
73 The reads are then separated based only on these loci. We did not find any variant caller suitable for our
74 specific challenge - calling variants with noisy long reads in a metagenomic context including potentially low-
75 abundance strains while maintaining high computational efficiency. Thus, we devised our own variant calling
76 procedure.

77

78 The naivest procedure to identify polymorphic loci consists in going through the pileup of the reads on the
79 assembly and identifying loci where at least a proportion p of reads have an alternative allele. However, this
80 approach falls short when using error-prone reads. For instance, in the case of a strain representing only 1%
81 of the total of the reads, p needs to be less than 0.01 to detect variant positions corresponding to this strain,
82 resulting in the selection of many artefactual positions if the reads have an error rate $> 1\%$.

83

84 The key lies in taking several loci into account simultaneously, an idea already explored in (Z Feng et al.,
85 2021) and leveraging the assumption that alignment artifacts occur randomly in the pileup while genomic
86 variant are expected to be correlated along the alignment. Consequently, pileups at polymorphic loci are ex-
87 pected to exhibit strong correlation, contrary to pileups at non-polymorphic loci. HairSplitter introduces a new
88 statistical approach and a new algorithm to exploit this observation and detect even rare strains, as illustrated

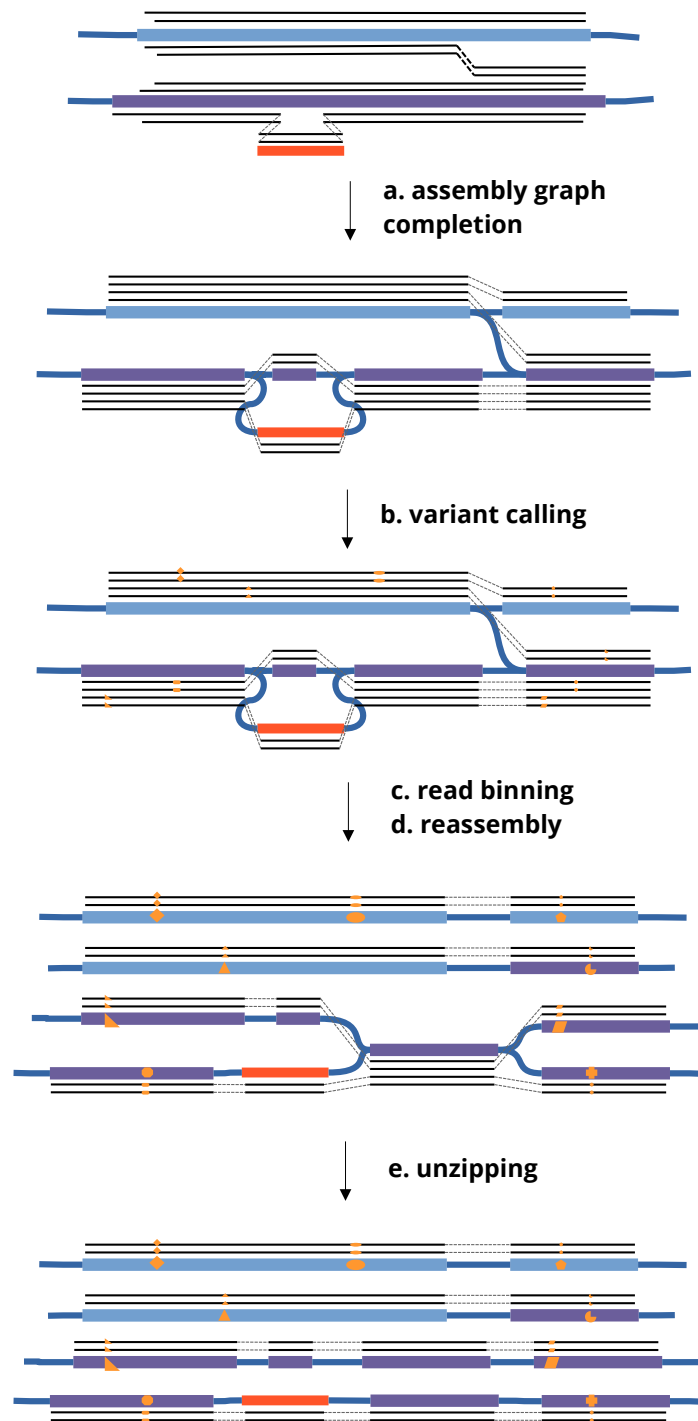


Figure 1. Illustration of the five steps of the HairSplitter pipeline. Colored rectangles represent contigs, thick blue lines are links in the assembly graph and black lines represent the reads aligned on the assembly. Orange shapes on reads and contigs indicate variant positions compared to the original sequence.

```

ref AACCAAGATAGACCAGATAGACACAGATTGGCGTTTAGGAACAGATGACAGATACGCA
r1 AACCAAGATAGA-CAGATAGACACAGATTGGCGTTTAGGAACAGATGACAGATA-GCA
r2 AACCAAGATAGAC-AGATAGCACAGGATTGGCGTTTAGGAACAGATGATAGATAC--A
r3 AACCAAGATAGA-CAGATAGACACAGATTGGCGTTTAGTAAACAGATGACAGATAGCCA
r4 AACCAAGATAGAC-AGATAGACACATATTGGCGTTTAGGAACATTTGACAGATA-GCA
r5 AACCAAGATAGA-CAGATAGGCACATATTGGCGTTTAGGAACAGTTGACAGATA--CGCA
r6 AACCAAGATAGAC-AGATAGACACATATTGGCGTTTAGGATCAGTTGACAGATA-GCA

```

Figure 2. In this pileup of reads, does the submatrix of variants highlighted in red vouch for the presence of two strains? The probability that there exist 3 reads having alternative allele at 3 loci if we estimate $e = 0.1$ is less than 0.02: the variants are thus likely not independent and probably underline the presence of at least two different strains.

89 below.

90

91 Consider a complete pileup of n reads over m positions, which we will model as a matrix of letters. Let us
 92 assume that errors occur independently on all reads and at all positions with a probability $\leq \epsilon$ and that all
 93 errors on a given column are identical (worst-case scenario). We aim to estimate the probability that there
 94 exist a reads that share errors at b different loci. In other words, the probability that there exist a submatrix
 95 of size $a * b$ containing only errors in the pileup, defined by selecting a rows (reads) and b columns (loci).

96 There exist $\binom{n}{a} \binom{m}{b}$ submatrices of size $a * b$. Each of these submatrix has probability lower than ϵ^{ab} to con-
 97 tain only errors. Therefore, given that the expectation is linear (DeGroot and Schervish, 2002), the expectation
 98 E of the number of submatrices of size $a * b$ containing only errors in the pileup is lower than $\binom{n}{a} \binom{m}{b} * \epsilon^{ab}$.
 99 Now, to obtain the probability that there exist no submatrix of size $a * b$ containing only errors, we can use
 100 Markov's inequality, according to which the probability that a positive random variable be higher than 1 is
 101 always smaller than the expectation of this variable (DeGroot and Schervish, 2002). Here, it tells us that the
 102 probability that there exist a submatrix containing only errors is smaller than E . In other terms, the probability
 103 that there exist somewhere in the pileup a reads sharing errors at b different loci is lower than $\binom{n}{a} \binom{m}{b} * \epsilon^{ab}$.

104 Now, let us consider a pileup with $n = 1000$ reads across $m = 5000$ positions and $\epsilon = 0.1$. The probability
 105 that there exist $a = 10$ reads sharing errors at $b = 10$ different loci is lower than $\binom{n}{a} \binom{m}{b} * \epsilon^{ab} = 9.10^{-44}$.
 106 Therefore, if the error rate is of 10% or less and the pileup indicates 10 reads (1% coverage) sharing an alter-
 107 native allele at 10 loci (divergence of 0.2%), we can confidently assume that these are not errors, suggesting
 108 these reads originate from the same strain, and the loci are polymorphic sites.

109

110 Despite its simplified nature, this model underscores the statistical power gained by examining multiple
 111 loci simultaneously, enabling the detection of low-abundance, highly similar strains even in the presence of
 112 very noisy long reads. The idea behind the model is illustrated in Figure 2.

113

114 Variant calling

115 The approach to identifying polymorphic loci capitalizes on the statistical power underlined above. Specifi-
 116 cally, HairSplitter aims to identify clusters of positions featuring alternative alleles on the same reads.

117

118 To generate the pileup, all reads are aligned to the contigs of the assembly using minimap2 (Li, 2018). Hair-
 119 Splitter then traverses the pileup of each contig and determines, for each position, the majority allele and the
 120 main alternative allele (either a base or an indel). Only positions with a minimum of five reads carrying alter-
 121 native alleles are considered potential polymorphic sites to ensure statistical robustness (cf. model above).
 122 HairSplitter compares each new position to previously observed positions. If the set of reads with alternative
 123 alleles at this position and at a previously encountered position share more than 90% reads, the new position
 124 is clustered with the old one.

125

126 After all positions have been considered, clusters are tested using the statistical model described above
127 and only clusters with a p-value below 0.001 are kept. The corresponding positions are outputted as polymor-
128 phic sites.

129

130 Read binning

131 The contig is divided into windows with a default size of w (2000 bases by default). Reads are binned by
132 haplotypes sequentially on the windows of a contig. Only reads spanning the entirety of the window are con-
133 sidered for binning. To cluster reads, HairSplitter operates on the premise that reads originating from the
134 same haplotype should be identical at all polymorphic loci. Nevertheless, inherent sequencing and variant-
135 calling errors might introduce unintended discrepancies among reads from a single haplotype. To address
136 this, HairSplitter adopts a three-step strategy.

137

138 Step one is to correct errors at polymorphic loci. HairSplitter corrects the errors at polymorphic loci by
139 performing a k-nearest-neighbour imputation (Fix and Hodges, 1989), with $k = 5$. The distance between two
140 reads is defined as the number of different alleles at polymorphic positions. Each base of the pileup is consid-
141 ered and changed to the most frequent base among the k nearest neighbours on all reads and all positions
142 until convergence.

143

144 Step two is to form clusters of reads, clustering reads together if and only if they exhibit no differences at
145 any polymorphic loci.

146

147 In the third step, a last check is run to rescue small clusters that can arise from errors in Step 1. HairSplitter
148 constructs a graph linking each read to its k closest neighbours, including links between all pairs of reads
149 differing on one position or less. The graph is then clustered using the Chinese Whispers algorithm (Biemann,
150 2006), initialising the clustering with the clusters obtained in the second step. The Chinese Whispers algorithm
151 iteratively assign reads to the most represented cluster among their neighbors until convergence. The Chi-
152 nese Whispers algorithm always converge toward a stable solution, i.e. a clustering where all reads are in the
153 same group as at least half of their neighbors. There exist many stable clusterings but the algorithm is likely
154 to converge to a solution close to the initialization: the clusters obtained in the second step are unlikely to be
155 significantly altered, but very small clusters will likely be merged with other close cluster.

156

157 Reassembly

158 Across all windows on every contig, the original sequence undergoes repolishing using the haplotype-
159 specific groups of reads previously identified. The polishing can be executed with either Racon (Vaser et al.,
160 2017) or Medaka (Medaka 2018), with the latter being more precise but considerably slower in our experience.
161 By default, HairSplitter uses Medaka exclusively for short genomes (≤ 1 Mb).

162 Graph Unzipping

163 The resulting assembly comprises contigs of length w that can easily be stitched into longer contigs. For
164 this purpose, a straightforward algorithm is employed, GraphUnzip (Faure et al., 2021), depicted in Figure 1e.
165 Let us call a contig exhibiting multiple outgoing links with other contigs at one end a “knot”. Knots generally
166 represent collapsed contigs. GraphUnzip initially aligns all reads on the assembly graph. Subsequently, Gra-
167 phUnzip iteratively assess nodes. If more than three reads traverse a neighbor of the knot (called A), then

dataset	species	# strains	strain coverages	ANI divergence	sequencing technology
HBV-2	hepatitis B	2	4000x, 9900x	10%	Nanopore R.9.4.1
Norovirus-7	Norovirus	7	50, 350, 450, 700, 900, 1150, 1400x	1-3.9 %	Nanopore R.9.4.1
<i>V. fluvialis</i>	<i>Vagococcus fluvialis</i>	5	90x, 136x, 172x, 182x, 206x	0.01-1.51%	Nanopore R9.4.1
Zymo-GMS Q9	<i>Escherichia coli</i>	5	90x, 90x, 90x, 90x, 90x	0.37-1.51%	Nanopore R9.4.1
Zymo-GMS Q20	<i>Escherichia coli</i>	5	25x, 25x, 25x, 25x, 25x	0.37-1.51%	Nanopore R10.4.1
Zymo-GMS HiFi	<i>Escherichia coli</i>	5	41x, 41x, 41x, 41x, 41x	0.37-1.51%	PacBio HiFi

Table 1. Characteristics of the different datasets used for benchmarking on real data.

168 traverse the knot, and traverse another neighbor at the opposite end of the knot (called B), the knot is du-
169 plicated to create a new contig which will have as unique neighbors A and B. The links from A and B to the
170 original knot are deleted, preserving only the links to the copy of the contig. This process is repeated until no
171 further knots can be duplicated.

172 Results

173 Datasets

174 The datasets used in this article are described in Table 1. The accession numbers of the data on public
175 repositories can be found in section* "Reproducibility and data availability".

176 Bacterial datasets

177 We used the Zymobiotics Gut Microbiome Standard (abbreviated to Zymo-GMS) and a *Vagococcus fluvialis*
178 dataset (Rodriguez Jimenez et al., 2022) to compare the performance of different algorithms designed to sepa-
179 rate bacterial haplotypes in a metagenomic context. Zymo-GMS is a mixture of bacteria, archaea and yeast, 21
180 different strains in total, dosed to mimic the composition of the human gut microbiome. These 21 strains in-
181 clude five *Escherichia coli* strains, which we used to evaluate the strain-separation ability of various programs.
182 Three Zymo-GMS sequencing were used, respectively from a Nanopore R9.4.1 run, a Nanopore 10.4.1 run
183 and a PacBio HiFi run. The *Vagococcus fluvialis* dataset consists of a mix of five *Vagococcus fluvialis* strains that
184 were sequenced together using barcoded reads, each barcode corresponding to a strain. We did not use the
185 barcode information for the assemblies, reserving them for validation. Among the five strains, three had an
186 ANI over 99.99%. metaFlye is used to assemble the reads, as it yielded better assemblies compared to Canu
187 according to Vicedomini et al. (Vicedomini et al., 2021).

188 In addition, we simulated datasets to assess the impact of the number of strains, coverage and divergence
189 on the assemblies. These experiments were directly inspired by the protocol of Vicedomini et al. (Vicedomini
190 et al., 2021). The genomes of ten strains of *Escherichia coli* were downloaded from the SRA, namely 12009
191 (GCA_000010745.1), IAI1 (GCA_000026265.1), F11 (GCA_018734065.1), S88 (GCA_000026285.2), Sakai (GCA_
192 003028755.1), SE15 (GCA_000010485.1), *Shigella flexneri* (GCF_000006925.2), UMN026 (GCA_000026325.2), HS
193 (GCA_000017765.1), and K12 (GCF_009832885.1). These strains were chosen to be representative of the diver-
194 sity of *E. coli*. We simulated Nanopore sequencing using Badread (R Wick, 2019) with the setting "Nanopore2023"
195 to simulate 50x of R10.4.1 reads. Between 2 and 10 strains were mixed to assess how many strains the soft-
196 ware could separate. From the 10-strain mix, the 12009 strain was downsampled to 30x, 20x, 10x and 5x
197 to assess the impact of the coverage on strain separation. Finally, to assess the impact of the divergence
198 of sequences on strain separation, 50x of reads were simulated for strain K12 and for strains of decreasing
199 divergence with K12; assemblies of K12 with each of these strain was evaluated for separation.

200 Viral datasets

201 Two datasets were used to benchmark the performance of the programs tested at separating viral haplo-
202 types, a 2-strain hepatitis B Virus (HBV) mix from (McNaughton et al., 2019) and an in-silico mix of the sequenc-

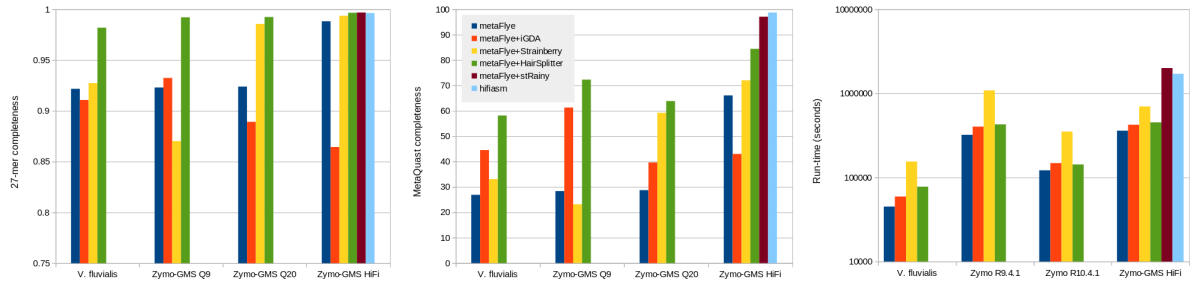


Figure 3. 27-mer completeness, MetaQUAST completeness and run-time of different software on the *Vagococcus* and the three Zymo-GMS dataset. The run-times are the run-times of the full assembly pipeline (assembly+strain separation) and are represented in log scale.

203 ing of seven strains of Norovirus from Cai et al. (Flint et al., 2021). These datasets were directly taken from
 204 the paper of HaploDMF (Cai et al., 2022). The reference genomes to run reference-based tools were taken as
 205 the reference genome in the GenBank database, GCF_000861825.2 for HBV and MW661279.1 for Norovirus.

206 Performance evaluation

207 We used MetaQUAST (Mikheenko et al., 2015) to measure assembly features such as assembly length,
 208 NG50, misassemblies, mismatches, indels and completeness. MetaQUAST was run with the `-unique-mapping`
 209 and `-reuse-combined-alignments` options to prevent a sequence, whether a contig or part of it, from being
 210 mapped to multiple distinct reference locations.

211 To assess if strains are well represented, the most important metric is the completeness of the resulting
 212 assembly. We chose to assess MetaQUAST completeness but also 27-mer completeness. MetaQUAST com-
 213 pleteness measures the percentage of the solution on which the assembly aligns, while 27-mer completeness
 214 measures the percentage of the 27-mers of the solution that are effectively found in the assembly. Collapsed
 215 homozygous contigs typically impact negatively MetaQUAST completeness but not 27-mer completeness.

216 Evaluated software

217 In addition of HairSplitter, we chose to evaluate the software stRainy (Kazantseva et al., 2023) and Strain-
 218 berry (Vicedomini et al., 2021), which have been introduced specifically as bacterial strain separation methods,
 219 hifiasm-meta (X Feng et al., 2022), which is the most popular assembler for direct HiFi assembly, Strainline (X
 220 Luo et al., 2022) and HaploDMF (Cai et al., 2022), which have been introduced as viral strain separation meth-
 221 ods and finally iGDA (Z Feng et al., 2021), which can perform both.

222 We have tried using all these software on all datasets. Strainline and HaploDMF failed to run in reasonable
 223 time on non-viral datasets and were automatically killed after 15 days of processing. Strainline failed to per-
 224 form strain separation on the HBV-2 dataset within its allowed RAM limit of 50G, probably because of the high
 225 coverage. We tried downsampling the dataset but the problem remained.

226 The reference-based virus phasing tools were run with the same reference genome as in (Cai et al., 2022),
 227 MT622522.1 for hepatitis B and MW661279.1 for Norovirus.

228 Benchmarking evaluation

229 Bacterial haplotypes

230 The benchmark results on the Zymo-GMS and *V. fluvialis* datasets are illustrated in Figure 3 and detailed
 231 in Supplementary Table 2. HairSplitter performed better separation of the conspecific strains compared to
 232 the original metaFlye assemblies, delivering more comprehensive and accurate assemblies than Strainberry

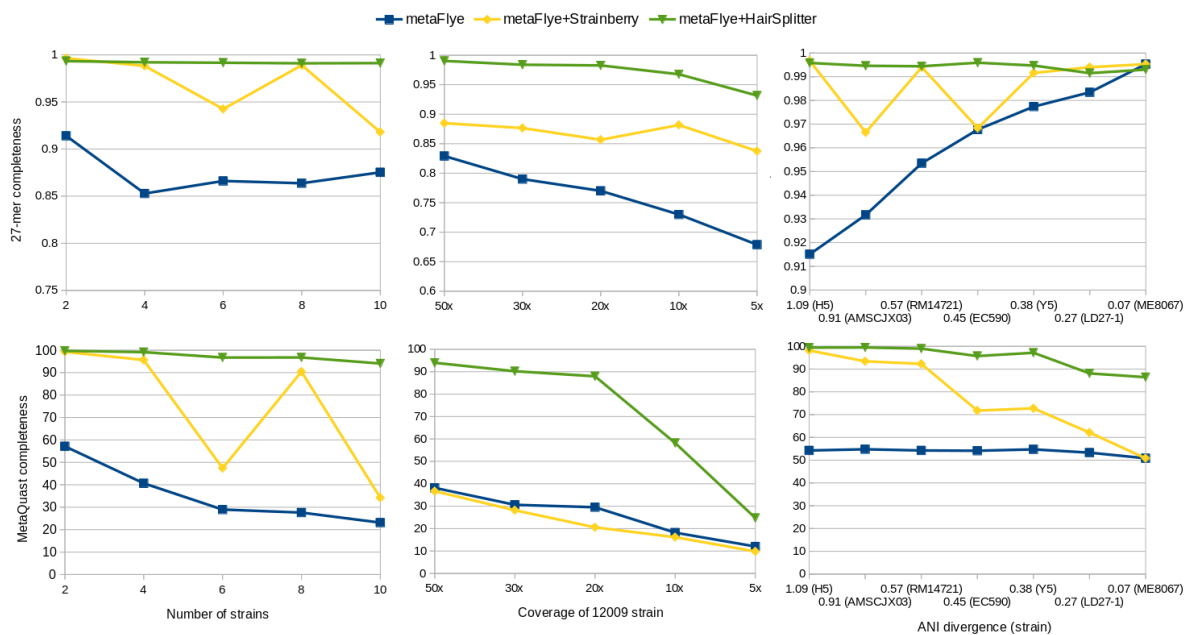


Figure 4. MetaQUAST completeness of assemblies of simulated metagenomes of *E. coli*. On the left, mix of 2 to 10 strains sequenced with 50x coverage were assembled. In the middle, strain 12009 was downsampled in the 10-strains metagenome and completeness of the 12009 strain is measured. On the right, reads of strains of decreasing divergence were mixed with K-12 reads and assembled.

233 and iGDA. Particularly with Nanopore data, HairSplitter produced the most complete assemblies, though less
 234 contiguous than those produced by Strainberry.

235 On HiFi reads, the stRainy, hifiasm and HairSplitter assemblies depicted a high k-mer completeness. How-
 236 ever, they showed either a high duplication ratio (for stRainy and hifiasm) or low metaQuast completeness
 237 (for HairSplitter) because none managed to duplicate repeated genomic regions to their correct multiplici-
 238 ties. This effect is also observed in several Nanopore assemblies, where 27-mer completeness remains high
 239 while MetaQUAST completeness is notably lower. Typically, the three almost identical *V. fluvialis* strains were
 240 assembled as one.

241 The completeness of assemblies in the simulated benchmark is presented in Figure 4, with a detailed evalu-
 242 ation in Supplementary Table 3. The evaluation of iGDA is not depicted because iGDA inexplicably decreased
 243 the completeness of the original metaFlye assemblies. Simulations indicated that HairSplitter significantly
 244 outperformed Strainberry, particularly in scenarios involving a high number of strains in the metagenome
 245 or highly similar strains. The relatively high completeness of the 8-strains Strainberry assembly can be at-
 246 tributed to its high duplication ratio. The completeness of HairSplitter assemblies decreased with the depth
 247 of coverage, especially below 20x coverage. The completeness also decreased slightly with the divergence
 248 of the strains, though the metaQuast completeness remained high (84%) when assembling two strains with
 249 0.07% divergence. Interestingly, the decline in MetaQUAST completeness with coverage and divergence was
 250 more pronounced than the decline in 27-mer completeness, highlighting HairSplitter's effectiveness in sepa-
 251 rating divergent regions and its difficulties in duplicating homozygous regions. This corresponds to the results
 252 observed in the Zymo-GMS datasets, where many pairwise divergences of strains were < 1%.

253 Viral haplotypes

254 The completeness results of the benchmark on the viral datasets are depicted Figure 5 and more complete
 255 evaluation of assemblies are available in Supplementary Table 5.

256 HaploDMF and HairSplitter managed to separate completely the HBV strains according to MetaQUAST.
 257 iGDA failed to recover the strains, while Strainberry outputted four different haplotypes instead of two (see

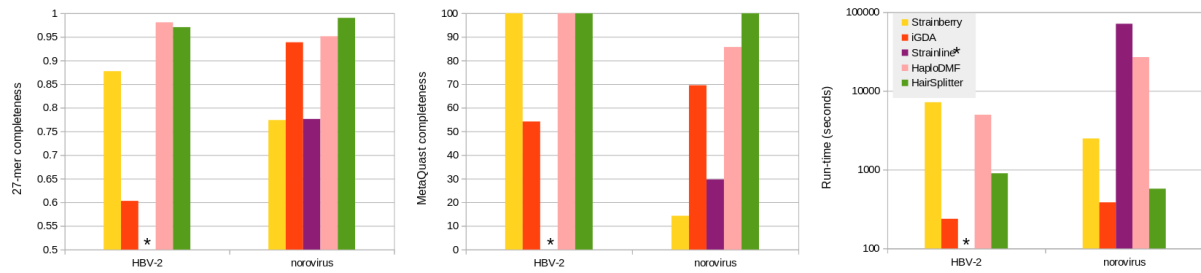


Figure 5. 27-mer completeness, MetaQUAST completeness and run-time of different software on the two viral datasets. Note that the run-time is shown in log scale. The Strainline assembly of HBV-2 is not shown because Strainline could not finish on this dataset.

258 supplementary Table 5). We checked that HaploDMF and HairSplitter separated the reads adequately, thus
 259 the slight differences in 27-mers completeness stem from polishing errors.

260 HairSplitter stood out as the sole software capable of successfully recovering all seven strains in the Norovirus
 261 mix, even capturing the least abundant strain comprising only 1% of the mix. To assess the sensitivity limits of
 262 HairSplitter in the viral context, we conducted two additional experiments within the Norovirus mix. In the first
 263 experiment, we decreased the relative abundance of the rarest strain to 0.1%, while maintaining 50x cover-
 264 age by uniformly increasing the coverage of the other strains. Remarkably, HairSplitter still achieved complete
 265 recovery (99.99% MetaQUAST completeness) of the rarest strain. The limited amount of data prevented us
 266 to further reduce the strain's relative abundance. In the second experiment, we uniformly diminished the
 267 coverage of all strains. The rarest strain was entirely recovered (99.99% MetaQUAST completeness) when cov-
 268 ered at $\geq 40x$, only the most divergent part of the virus was recovered (26.4% MetaQUAST completeness) at
 269 coverage 20x and 30x, and the strain was not recovered at all at 10x coverage. The primary determinant of
 270 HairSplitter's sensitivity thus seems to be absolute coverage rather than the strain's relative coverage.

271 Discussion

272 In this manuscript, we introduce HairSplitter, a pipeline to assemble haplotypes separately using an input
 273 assembly and long reads. The pipeline includes two main novelties, a program that completes an assembly
 274 graph and a read separation procedure. HairSplitter proved useful when dealing with noisy data ($\geq 1\%$ error
 275 rate), whereas its usefulness on HiFi reads compared to specialised software such as hifiasm or stRainy is de-
 276 batable. We show that HairSplitter can effectively separate several highly similar strains in both bacterial and
 277 viral contexts. Compared to the state of the art, HairSplitter can deal with a higher number of strains, lower
 278 relative abundances and lower strain divergence, while maintaining a low computational cost.

279
 280 HairSplitter encounters a major limitation when strains have many homozygous regions. In these regions,
 281 it is not possible to assign reads to specific haplotype groups, making it necessary to duplicate the homozy-
 282 gous regions to their correct multiplicity in order to fully recover the strains. This study has demonstrated
 283 that this is a challenging problem that current assemblers have not been able to successfully address in the
 284 HiFi dataset. Further investigation is needed to solve this issue. A lead could be to use astutely the topology
 285 of the assembly graph.

286
 287 A direction for future work would also be to generalize the assembly graph completion module. The idea
 288 of the module is to make sure all reads align end-to-end onto the assembly graph. We believe such a module
 289 could be useful to improve many assemblies. However, the version implemented for now in HairSplitter is
 290 very basic and does not perform well in repeated, complicated regions of the graph. A more sophisticated
 291 module could involve local reassembly and iterative graph completion.

293 Since HairSplitter is already successful at separating both bacterial and viral haplotypes, we expect to be
294 able to extend this work naturally towards the phasing of polyploid organisms, including highly heterozygous
295 non-model organisms, which remains an open problem (Guiglielmoni et al., 2021). For this particular case,
296 some extra information could be leveraged to improve the HairSplitter pipeline, such as the fact that all hap-
297 lotypes are expected to be equally abundant and that the total number of haplotype is usually known.

298 **Reproducibility and data availability**

299 The HairSplitter code can be found on github at <https://github.com/rolandfaure/hairsplitter>.

300 The experiments were run with Flye 2.9.2-b1786, hifiasm HairSplitter v1.9.4, HaploDMF commit a07d082c3,
301 Strainline commit 8d26341, iGDA commit 54ecec9, Strainberry v1.1, stRainy commit 34573cd, hifiasm-meta
302 v0.3-r063.2, minimap2 v2.26-r1175 and Quast v5.2.0.

303 HBV sequencing reads can be found under accession number ERR3253560 in SRA. The seven Norovirus sets
304 of reads can be found under accession numbers SRR13951181, SRR13951181, SRR13951186, SRR13951185,
305 SRR13951184, SRR13951165 and SRR13951160. The *Vagococcus fluvialis* data are accessible under project
306 PRJNA755170 in SRA. The Zymo-GMS sequencing data can be found under accession numbers SRR17913200,
307 SRR17913199 and SRR13128013.

308 All the assemblies, simulated data and command lines used are available on Zenodo, DOI 10.5281/zen-
309 odo.10495033, <https://zenodo.org/records/11639887>.

310 **Acknowledgments**

311 We thank Ulysse Faure for his mathematical help. Alexandros Vasilikopoulos, Andrew Woodruff and Alessan-
312 dro Derzelle tested HairSplitter and kindly helped debugging.

313 We acknowledge the GenOuest bioinformatics core facility (<https://www.genouest.org>) for providing the
314 computing infrastructure. The programs Tablet (Milne et al., 2013) and Bandage (RR Wick et al., 2015) were
315 used to visualize data while developing HairSplitter.

316 For the purpose of Open Access, a CC-BY public copyright licence has been applied by the authors to the
317 present document and will be applied to all subsequent versions up to the Author Accepted Manuscript arising
318 from this submission

319 **Fundings**

320 This work was funded by a Ph.D. AMX grant.

321 **Conflict of interest disclosure**

322 The authors declare that they comply with the PCI rule of having no financial conflicts of interest in relation
323 to the content of the article. The authors declare the following non-financial conflict of interest: Jean-François
324 Flot is a recommender of PCI Genomics.

325 **References**

326 Bertrand D, J Shaw, M Kalathiyappan, AHQ Ng, MS Kumar, C Li, M Dvornicic, JP Soldo, JY Koh, C Tong, OT Ng,
327 T Barkham, B Young, K Marimuthu, KR Chng, M Sikic, and N Nagarajan (Aug. 2019). Hybrid metagenomic
328 assembly enables high-resolution analysis of resistance determinants and mobile elements in human mi-
329 crobiomes. en. *Nature Biotechnology* 37, 937–944. ISSN: 1087-0156, 1546-1696. <https://doi.org/10.1038/s41587-019-0191-2>.

331 Biemann C (July 2006). Chinese whispers: An efficient graph clustering algorithm and its application to natural
332 language processing problems. *Proceedings of TextGraphs*, 73–80.

333 Cai D, J Shang, and Y Sun (Oct. 2022). HaploDMF: viral Haplotype reconstruction from long reads via Deep
334 Matrix Factorization. *Bioinformatics* 38. <https://doi.org/10.1093/bioinformatics/btac708>.

335 Ceppellini R, E Curtoni, P Mattiuz, V Miggiano, G Scudeller, and A Serra (1967). Genetics of leukocyte antigens:
336 a family study of segregation and linkage. In: *Report of Histocompatibility testing 1967*. Ed. by Curtoni E.S.
337 Mattiuz P.L. TR.

338 Cesare Md, M Chimfwembe, A Jeffreys, J Chirwa, C Drakeley, K Schneider, B Mambwe, K Glanz, C Ntalla, M
339 Carrasquilla, S Portugal, R Verity, J Bailey, I Ghinai, G Busby, B Hamainza, M Hawela, D Bridges, and J
340 Hendry (Feb. 2024). Flexible and cost-effective genomic surveillance of *P. falciparum* malaria with targeted
341 nanopore sequencing. *Nature Communications* 15. <https://doi.org/10.1038/s41467-024-45688-z>.

342 Cheng H, G Concepcion, X Feng, H Zhang, and H Li (Feb. 2021). Haplotype-resolved de novo assembly using
343 phased assembly graphs with hifiasm. *Nature Methods* 18, 1–6. <https://doi.org/10.1038/s41592-020-01056-5>.

344

345 Coban O, G Deyn, and M Ploeg (Mar. 2022). Soil microbiota as game-changers in restoration of degraded lands.
346 *Science* 375, abe0725. <https://doi.org/10.1126/science.abe0725>.

347 Conlon M and A Bird (Dec. 2014). The Impact of Diet and Lifestyle on Gut Microbiota and Human Health.
348 *Nutrients* 7, 17–44. <https://doi.org/10.3390/nu7010017>.

349 DeGroot M and M Schervish (Jan. 2002). *Probability and Statistics*. Pearson. ISBN: ISBN 978-0-321-50046-5.

350 Faure R, N Guiglielmoni, and JF Flot (Feb. 2021). GraphUnzip: unzipping assembly graphs with long reads and
351 Hi-C. *bioRxiv*. <https://doi.org/10.1101/2021.01.29.428779>.

352 Feng X, H Cheng, D Portik, and H Li (June 2022). Metagenome assembly of high-fidelity long reads with hifiasm-
353 meta. *Nature Methods* 19, 1–4. <https://doi.org/10.1038/s41592-022-01478-3>.

354 Feng Z, J Clemente, B Wong, and E Schadt (May 2021). Detecting and phasing minor single-nucleotide variants
355 from long-read sequencing data. *Nature Communications* 12, 3032. <https://doi.org/10.1038/s41467-021-23289-4>.

356

357 Fix E and JL Hodges (1989). Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties.
358 *International Statistical Review* 57, 238–247. ISSN: 03067734, 17515823.

359 Flint A, S Reaume, J Harlow, E Hoover, K Weedmark, and N Nasheri (Sept. 2021). Genomic Analysis of Human
360 Noroviruses Using Combined Illumina-Nanopore Data. *Virus Evolution* 7. <https://doi.org/10.1093/ve/veab079>.

361

362 Frank C, D Werber, JP Cramer, M Askar, M Faber, M an der Heiden, H Bernard, A Fruth, R Prager, A Spode,
363 M Wadl, A Zoufaly, S Jordan, MJ Kemper, P Follin, L Müller, LA King, B Rosner, U Buchholz, K Stark, and G
364 Krause (2011). Epidemic Profile of Shiga-Toxin–Producing *Escherichia coli* O104:H4 Outbreak in Germany.
365 *New England Journal of Medicine* 365, 1771–1780. <https://doi.org/10.1056/NEJMoa1106483>.

366 Ghurye J, V Cepeda-Espinoza, and M Pop (Sept. 2016). Metagenomic Assembly: Overview, Challenges and
367 Applications. *The Yale Journal of Biology and Medicine* 89, 353–362.

368 Guiglielmoni N, A Houtain, A Derzelle, K Doninck, and JF Flot (June 2021). Overcoming uncollapsed haplotypes
369 in long-read assemblies of non-model organisms. *BMC Bioinformatics* 22. <https://doi.org/10.1186/s12859-021-04118-3>.

370

371 Kang X, X Luo, and A Schönhuth (Sept. 2022). StrainXpress: strain aware metagenome assembly from short
372 reads. en. *Nucleic Acids Research* 50, e101–e101. ISSN: 0305-1048, 1362-4962. <https://doi.org/10.1093/nar/gkac543>.

373

374 Kazantseva E, A Donmez, M Pop, and M Kolmogorov (Feb. 2023). *stRainy: assembly-based metagenomic strain*
375 *phasing using long reads*. en. preprint. Bioinformatics. <https://doi.org/10.1101/2023.01.31.526521>.

376 Kolmogorov M, DM Bickhart, B Behsaz, A Gurevich, M Rayko, SB Shin, K Kuhn, J Yuan, E Pevnikov, TPL Smith,
377 and PA Pevzner (Nov. 2020). metaFlye: scalable long-read metagenome assembly using repeat graphs. en.
378 *Nature Methods* 17, 1103–1110. ISSN: 1548-7091, 1548-7105. <https://doi.org/10.1038/s41592-020-00971-x>.

379 Konstantinidis K and J Tiedje (Mar. 2005). Genomic insights that advance the species definition for prokaryotes.
380 *Proceedings of the National Academy of Sciences of the United States of America* 102, 2567–72. [https://doi.org/](https://doi.org/10.1073/pnas.0409727102)
381 [10.1073/pnas.0409727102](https://doi.org/10.1073/pnas.0409727102).

382 Koren S, BP Walenz, K Berlin, JR Miller, NH Bergman, and AM Phillippy (May 2017). Canu: scalable and accurate
383 long-read assembly via adaptive *k*-mer weighting and repeat separation. en. *Genome Research* 27, 722–
384 736. ISSN: 1088-9051, 1549-5469. <https://doi.org/10.1101/gr.215087.116>.

385 Li H (Sept. 2018). Minimap2: pairwise alignment for nucleotide sequences. en. *Bioinformatics* 34. Ed. by Birol I,
386 3094–3100. ISSN: 1367-4803, 1367-4811. <https://doi.org/10.1093/bioinformatics/bty191>.

387 Li H, X Feng, and C Chu (Oct. 2020). The design and construction of reference pangenome graphs with mini-
388 graph. *Genome Biology* 21, 265. <https://doi.org/10.1186/s13059-020-02168-z>.

389 Luo C, R Knight, H Siljander, M Knip, R Xavier, and D Gevers (Sept. 2015). ConStrains identifies microbial strains
390 in metagenomic datasets. *Nature biotechnology* 33. <https://doi.org/10.1038/nbt.3319>.

391 Luo R and Y Lin (2023). VStrains: De Novo Reconstruction of Viral Strains via Iterative Path Extraction from As-
392 sembly Graphs. In: *Research in Computational Molecular Biology*. Ed. by Tang H. Cham: Springer Nature
393 Switzerland, pp. 3–20. ISBN: 978-3-031-29119-7.

394 Luo X, X Kang, and A Schönhuth (Jan. 2022). Strainline: full-length de novo viral haplotype reconstruction from
395 noisy long reads. *Genome Biology* 23. <https://doi.org/10.1186/s13059-021-02587-6>.

396 Magazine N, T Zhang, Y Wu, M McGee, G Veggiani, and W Huang (Mar. 2022). Mutations and Evolution of the
397 SARS-CoV-2 Spike Protein. *Viruses* 14, 640. <https://doi.org/10.3390/v14030640>.

398 McNaughton A, H Roberts, D Bonsall, Md Cesare, J Mokaya, S Lumley, T Golubchik, P Piazza, J Martin, C Lara,
399 A Brown, M Ansari, R Bowden, E Barnes, and P Matthews (May 2019). Illumina and Nanopore methods for
400 whole genome sequencing of hepatitis B virus (HBV). *Scientific Reports* 9. [https://doi.org/10.1038/s41598-](https://doi.org/10.1038/s41598-019-43524-9)
401 [019-43524-9](https://doi.org/10.1038/s41598-019-43524-9).

402 Medaka (2018). github.com/nanoporetech/medaka.

403 Mikheenko A, V Saveliev, and A Gurevich (Nov. 2015). MetaQUAST: Evaluation of metagenome assemblies.
404 *Bioinformatics* 32, btv697. <https://doi.org/10.1093/bioinformatics/btv697>.

405 Milne I, G Stephen, M Bayer, PJA Cock, L Pritchard, L Cardle, PD Shaw, and D Marshall (Mar. 2013). Using Tablet
406 for visual exploration of second-generation sequencing data. en. *Briefings in Bioinformatics* 14, 193–202.
407 ISSN: 1467-5463, 1477-4054. <https://doi.org/10.1093/bib/bbs012>.

408 Quince C, S Nurk, S Raguideau, R James, OS Soyer, JK Summers, A Limasset, AM Eren, R Chikhi, and AE Darling
409 (Sept. 2020). *Metagenomics Strain Resolution on Assembly Graphs*. en. preprint. Bioinformatics. [https://doi.](https://doi.org/10.1101/2020.09.06.284828)
410 [org/10.1101/2020.09.06.284828](https://doi.org/10.1101/2020.09.06.284828).

411 Rodriguez Jimenez A, N Guiglielmoni, L Goetghebuer, E Dechamps, I George, and JF Flot (Aug. 2022). Com-
412 parative genome analysis of *Vagococcus fluvialis* reveals abundance of mobile genetic elements in sponge-
413 isolated strains. *BMC Genomics* 23. <https://doi.org/10.1186/s12864-022-08842-9>.

414 Runtuwene L, J Tuda, A Mongan, and Y Suzuki (Apr. 2019). On-Site MinION Sequencing. In: pp. 143–150. ISBN:
415 978-981-13-6036-7. https://doi.org/10.1007/978-981-13-6037-4_10.

416 Vaser R, I Sovic, N Nagarajan, and M Sikic (Jan. 2017). Fast and accurate de novo genome assembly from long
417 uncorrected reads. *Genome Research* 27, gr.214270.116. <https://doi.org/10.1101/gr.214270.116>.

418 Vicedomini R, C Quince, AE Darling, and R Chikhi (July 2021). Strainberry: automated strain separation in low-
419 complexity metagenomes using long reads. en. *Nature Communications* 12, 4485. ISSN: 2041-1723. [https:](https://doi.org/10.1038/s41467-021-24515-9)
420 [//doi.org/10.1038/s41467-021-24515-9](https://doi.org/10.1038/s41467-021-24515-9).

421 Ward N (Apr. 2006). New directions and interactions in metagenomics research. *FEMS microbiology ecology* 55,
422 331–8. <https://doi.org/10.1111/j.1574-6941.2005.00055.x>.

423 Wick R (Apr. 2019). Badread: simulation of error-prone long reads. *Journal of Open Source Software* 4, 1316.
424 ISSN: 2475-9066. <https://doi.org/10.21105/joss.01316>.

425 Wick RR, MB Schultz, J Zobel, and KE Holt (Oct. 2015). Bandage: interactive visualization of *de novo* genome
426 assemblies. en. *Bioinformatics* 31, 3350–3352. ISSN: 1367-4811, 1367-4803. [https://doi.org/10.1093/](https://doi.org/10.1093/bioinformatics/btv383)
427 [bioinformatics/btv383](https://doi.org/10.1093/bioinformatics/btv383).

Supplementary material

		Completeness (%)	Duplication ratio	NGA50	#misassemblies	#mismatches per 100 kbp	# indels per 100 kbp	Assembly length (Mb)
<i>Vagococcus fluvialis</i>	metaFlye	26.90	1.097	-	40	340.14	383.57	4.4
	metaFlye + iGDA	44.530	1.151	1313	2	115.57	391.70	7.4
	metaFlye + Strainberry	33.112	1.119	-	45	77.71	510.25	5.5
	metaFlye + HairSplitter	58.150	1.066	19881	31	102.49	410.95	9.1
Zymo-GMS Q9*	metaFlye	28.365	1.048	-	66	286.86	38.80	7.4
	metaFlye + iGDA	61.293	1.489	12450	21	225.67	41.30	22.6
	metaFlye + Strainberry	23.166	1.072	-	45	191.51	51.65	6.1
	metaFlye + HairSplitter	72.293	1.105	9974	24	65.77	39.60	19.8
Zymo-GMS Q20*	metaFlye	28.742	1.051	-	62	300.25	34.41	7.5
	metaFlye + iGDA	39.650	1.118	-	8	181.55	28.23	11.0
	metaFlye + Strainberry	59.197	1.138	28421	66	193.55	42.42	16.7
	metaFlye + HairSplitter	63.837	1.022	12000	43	40.01	21.69	16.2
Zymo-GMS HiFi*	metaFlye	66.064	1.076	79832	39	92.55	6.65	17.6
	metaFlye + iGDA	42.996	1.515	17104	15	102.70	9.96	16.1
	metaFlye + Strainberry	72.016	1.142	53249	46	57.53	6.92	20.4
	metaFlye + HairSplitter	84.418	1.286	25851	69	32.35	12.16	26.9
	metaFlye + stRainy	97.078	1.737	41195	47	44.15	12.26	41.8
	hifiasm	98.732	1.911	288422	82	30.07	4.99	46.7

Table 2. metaQuast metrics of the bacterial assemblies obtained from experimental data. *metrics are computed with respect to the 5 *E. coli* strains, not the complete dataset - the assembly length is the aligned assembly length on the *E. coli* reference

		Completeness (%)	Duplication ratio	NGA50	#misassemblies	#mismatches per 100 kbp	# indels per 100 kbp	Assembly length (Mb)
# strains								
2	metaFlye	57.137	1.043	61559	22	216.18	216.53	6.1
	metaFlye + Strainberry	99.268	1.074	701492	11	24.83	64.73	10.8
	metaFlye + HairSplitter	99.716	1.008	396746	1	8.21	42.73	10.2
4	metaFlye	40.666	1.071	-	50	562.91	268.83	8.9
	metaFlye + Strainberry	95.631	1.148	251144	39	93.07	81.88	22.3
	metaFlye + HairSplitter	99.109	1.064	109320	39	25.74	50.83	21.3
6	metaFlye	28.949	1.087	-	63	585.20	264.89	9.4
	metaFlye + Strainberry	47.430	1.108	8151	90	289.19	92.23	15.7
	metaFlye + HairSplitter	96.717	1.086	77500	125	52.55	56.23	31.1
8	metaFlye	27.599	1.051	-	76	527.44	277.85	11.5
	metaFlye + Strainberry	90.438	1.533	83755	157	179.31	172.25	54.7
	metaFlye + HairSplitter	96.759	1.180	45253	244	87.74	84.97	45.1
10	metaFlye	23.130	1.036	-	79	469.27	277.71	11.7
	metaFlye + Strainberry	34.207	1.095	-	175	363.06	137.01	18.2
	metaFlye + HairSplitter	94.045	1.192	41223	262	93.74	66.47	54.5
coverage								
30x*								
	metaFlye	30.618	1.032	-	1	719.80	55.27	1.7
	metaFlye + Strainberry	28.188	1.085	-	2	347.21	127.06	1.7
	metaFlye + HairSplitter	90.206	1.170	40243	19	143.08	41.63	5.8
20x*								
	metaFlye	29.522	1.018	-	6	859.11	76.03	1.7
	metaFlye + Strainberry	20.562	1.037	-	3	274.86	102.01	1.2
	metaFlye + HairSplitter	87.948	1.093	37879	22	130.95	77.82	5.2
10x*								
	metaFlye	18.201	1.005	-	1	498.17	85.10	1.0
	metaFlye + Strainberry	16.178	1.007	-	1	347.06	181.70	0.9
	metaFlye + HairSplitter	58.172	1.054	10763	6	214.47	117.59	3.3
5x*								
	metaFlye	12.020	1.010	-	2	849.88	201.29	0.6
	metaFlye + Strainberry	9.807	1.013	-	2	422.61	273.49	0.5
	metaFlye + HairSplitter	24.746	1.082	-	2	464.50	197.55	1.5
divergence								
H5								
(1.09%)	metaFlye	54.246	1.002	19206	22	324.97	29.23	5.1
	metaFlye + Strainberry	98.157	1.001	652572	2	324.97	1.35	9.3
	metaFlye + HairSplitter	99.419	1.008	294365	6	8.08	15.25	9.4
AMSCJX03								
(0.91%)	metaFlye	54.783	1.001	18675	17	254.48	28.42	5.0
	metaFlye + Strainberry	93.390	1.003	279448	3	0.73	1.98	8.6
	metaFlye + HairSplitter	99.456	1.002	387661	11	10.96	23.29	9.2
RM74721								
(0.57%)	metaFlye	54.254	1.000	19360	19	132.60	11.73	5.0
	metaFlye + Strainberry	92.256	1.006	380826	1	2.97	8.57	8.6
	metaFlye + HairSplitter	98.957	1.007	265482	14	15.54	29.08	9.2
EC590								
(0.45%)	metaFlye	54.132	1.000	17337	10	117.79	12.85	5.0
	metaFlye + Strainberry	71.749	1.003	156627	10	9.29	1.72	6.6
	metaFlye + HairSplitter	95.697	1.024	190750	6	6.46	24.51	9.0
Y5								
(0.38%)	metaFlye	54.736	1.002	22104	21	63.85	9.38	5.2
	metaFlye + Strainberry	72.758	1.006	181387	13	8.64	3.28	6.9
	metaFlye + HairSplitter	97.154	1.024	253619	8	40.80	52.36	9.4
LD27-1								
(0.27%)	metaFlye	53.295	1.001	19411	8	43.83	5.33	5.0
	metaFlye + Strainberry	62.101	1.004	112749	13	7.41	3.53	5.8
	metaFlye + HairSplitter	88.101	1.055	137245	2	95.36	97.47	8.6
ME8067								
(0.07%)	metaFlye	50.820	1.000	47356	8	10.29	3.19	4.7
	metaFlye + Strainberry	50.820	1.000	47356	8	10.29	3.19	4.7
	metaFlye + HairSplitter	86.419	1.064	131660	0	84.75	80.83	8.5

Table 3. metaQuast metrics of the bacterial assemblies obtained from simulated Nanopore R10.4.1 data. * The metrics displayed for the downsampled datasets are the metrics computed with respect to the downsampled strain, and not with respect to the complete 10 strains.

		Completeness (%)	Duplication ratio	NGA50	#misassemblies	# mismatches per 100 kbp	# indels per 100 kbp
HBV-2	Strainberry	99.984	2.174	4504	3	881.59	1562.50
	iGDA	54.174	1.001	1081	0	201.15	229.89
	Strainline						
	HaploDMF	99.984	1.000	3207	0	15.58	93.46
	HairSplitter	99.953	1.001	3209	0	46.72	109.02
norovirus	Strainberry	14.283	1.000	-	0	52.97	13.24
	iGDA	69.514	1.548	2838	0	112.55	15.83
	Strainline	29.659	5.787	7541	0	479.44	136.67
	HaploDMF	85.702	1.000	7549	0	165.60	26.50
	HairSplitter	100.000	1.038	7550	0	107.57	35.96

Table 4. metaQuast metrics of the viral assemblies.

Number of strains		metaFlye assembly	metaFlye assembly after graph completion
2	N50	374204	374204
	#misassemblies	22	20
4	N50	136064	47179
	#misassemblies	50	17
6	N50	78959	36209
	#misassemblies	63	12
8	N50	73412	32985
	#misassemblies	76	18
10	N50	55801	26906
	#misassemblies	79	19

Table 5. N50 and metaQuast-measured number of misassemblies of simulated datasets with varying number of *E. coli* strains, before and after completing the assembly graph. Since the completion step breaks contigs, the N50 diminishes. The number of misassemblies diminishes with graph completion.