Chapter II: Microbial Genome Reconstruction from Ancient DNA- Assembly Methods

Luis A. Arriola
October 2017

Comparative assessment of methods for reconstruction of whole genomic sequences from ancient microbial organisms

INTRODUCTION

In recent years significant technological and analytical advances have revolutionized the study of ancient microbial organisms. High-throughput sequencing technologies, such as Illumina NGS and enrichment capture, have improved the recovery of endogenous DNA sequences of microorganisms hundreds to thousands of years old (Bos *et al.*, 2011, 2016; Feldman *et al.*, 2016; Hofreiter *et al.*, 2015). However, few progress has been made in the tools used to reconstruct and analyze whole genomic sequences of these organisms (Hofreiter *et al.*, 2015; Parks and Lambert, 2015; Schubert *et al.*, 2012). While single or multi-locus analyses offer a convenient approach to identify and classify organisms, they offer reduced power to resolve phylogenetic relationships, study the evolution and biology of an organism, and in some cases can lead to misleading results due to its low resolution (REF). On the other hand, genome-wide information recovered from historical sources provides important insights into the biology and evolution of ancient organisms (REF) as it allows researchers to identify organisms unambiguously, refine models, and test hypothesis created from current genetic or genomic data at a higher resolution (REF). Differences between genomes from the same species or closely related species are used to establish more detailed phylogenetic relations, calculate mutation rates, and identify patterns of adaptation and evolution over time at the individual and population level (Parks *et al.*, 2015). Moreover, dated ancient genomic information adds a temporal dimension to genomic variation and provides details of the geographic distribution of the organisms in the past (REF).

The reconstruction of a genomic sequence is a critical step in the study of an organism and its evolution since all downstream analyses, comparisons and inferences depend and will be affected by the quality and fidelity of it (Giese *et al.*, 2014; Leonardi *et al.*, 2016; Schubert *et al.*, 2012). Currently, two main approaches are used to reconstruct whole microbial genomes from high-throughput sequencing (HTS) data: 'mapping assembly', which uses an already available reference sequence as a guide to locate and align fragments to their most likely locations; and 'de novo assembly', which is a reference-free reconstruction that matches overlapping sequences to create longer contiguous DNA reads (contigs) (Hofreiter *et al.*, 2015). In palaeomicrobiology (i.e., the study of ancient microbial organisms), mapping assembly is the most widely used approach for whole genome reconstruction, and researchers have been able to reconstruct draft genomic sequences from a variety of ancient microbial pathogens at various points in the past, including *Yersinia pestis* (Bos *et al.*, 2011, 2016; Feldman *et al.*, 2016; Rasmussen *et al.*, 2015; Wagner *et al.*, 2014), *Mycobacterium leprae* (Mendum *et al.*, 2014; Schuenemann *et al.*, 2013), *Mycobacterium tuberculosis* (Bos *et al.*, 2014; Kay *et al.*, 2015), *Helicobacter pylori* (Maixner *et al.*, 2016), *Treponema denticola* (Maixner *et al.*, 2014), *Brucella melitensis* (Kay *et al.*, 2014), and *Methanobrevibacter oralis subsp. neandertalensis* (Weyrich *et al.*, 2017). The reference mapping assembly approach is advantageous because it works well with low abundance and short fragment lengths characteristic of aDNA (Dabney *et al.*, 2013; Schubert *et al.*, 2012), and it allows the recovery of endogenous sequences in a complex system (i.e., metagenomic shotgun sequence datasets) (Maixner *et al.*, 2016). However, due to the stringency of this approach, it recovers highly conserved regions of the genome, and this hinders the ability to find regions that have greatly changed through time between the target organism and the used reference (Hofreiter *et al.*, 2015). Moreover, the amount of sequence recovered is highly dependent on the availability, quality, and choice of the sequence used as a reference, and the divergence and the heterogeneity of the target organism. Mapping assembly algorithms are affected by DNA damage patterns which result in lower yields of coverage, as reads are mapped with lower confidence, and in some cases allow reads to artificially map to other genomic locations with similar composition or the introduction of non-endogenous reads (Schubert *et al.*, 2012). The most widely used algorithm in aDNA studies is the Burrows-Wheelers Aligner (BWA), which specializes in short fragment mapping (Li and Durbin, 2009; Orlando *et al.*, 2015).

In 2012, Schubert *et al.* explored a series of parameters to improve the mapping of aDNA reads to modern reference genomes. In this survey, they showed that by deactivating the use of a seed region and increasing tolerance to higher edit distances, it was possible to increase the number of high-quality endogenous hits recovered, albeit in low proportions for Illumina data (Schubert *et al.*, 2012). Since then, these findings have been used by the aDNA community as the gold standard for mapping assembly. In contrast, the de novo assembly approach has only been successful in reconstructing an ancient microbial genome under very special circumstances (Schuenemann *et al.*, 2013, [and Yersinia pestis]). As this method does not depend on a reference sequence, it allows (in theory) a less biased reconstruction. The main disadvantage of the method is that it is highly dependant on the quality of the dataset, as the low fragment length and the presence of non-endogenous sequences complicate a de novo reconstruction (Westbury *et al.*, 2017). In addition to reference-based mapping and de novo assembly, other methods have been developed to assemble small genomes. These include reference guided assembly, which makes use of the advantages of de novo assembly to reconstruct contigs that could contain highly variable regions, and known reference sequences as a scaffold to locate the most likely positions for these contigs (Rajaraman *et al.*, 2013). Another novel method is iterative mapping. This method uses an initial mapping or de novo step to create self-standing reference sequences that will be cyclically extended via a reference mapping approach until no more new reads can be added (REF). This method has been used successfully to reconstruct mitochondrial sequences, with the advantage of not having to rely on a full genomic reference sequence and be able to work on complex datasets (metagenomics) (Hahn *et al.*, 2013; Westbury *et al.*, 2017).

Reconstructing microbial genomes presents certain advantages, as they count with small haploid genomes, which allow for reduced mapping/assembly times compared to higher organisms; however, some disadvantages also need to be considered. Ancient bacterial genome reconstruction represents a challenge due to lower yields of coverage derived from post-mortem damage. Other complications include genetic heterogeneity, a higher genetic divergence from modern organisms, lateral gene transfer, lower presence in samples and contamination. Many different strategies have been used in palaeomicrobiology studies to reconstruct whole genome information (Table 1). Nevertheless, to our knowledge, no formal comparison of mapping algorithms other than BWA has been conducted, nor has the impact of post-mortem aDNA damage on the reconstruction of genomes from microorganisms been investigated. Therefore it is essential to explore how different conditions found on aDNA, such as short read sizes, DNA damage, and genomic divergence, affect the mappability of aDNA reads and the reconstruction of microbial genomes. In this study, we compare three different approaches for whole-genome reconstruction of microbial genomes on three simulated datasets showing ancient DNA damage profiles and characteristics. We identified biases presented by each approach and propose possible ways to avoid them. Finally, we apply these approaches to real data to provide X% better genome coverage of Methanobrevibacter oralis neandertalensis, the oldest microbial ancient genome reported to date, reconstructed using conventional methods.

Table 01. Whole genome reconstruction approaches used in palaeomicrobiology.

METHODS

Simulated Datasets

In this study we used three simulated datasets to evaluate different strategies used for whole genome reconstruction of ancient microbial organisms. These datasets contained reads simulating different conditions observed on ancient DNA data, such as genomic divergence to reference sequence, aDNA damage profiles, and contamination context.

Divergence Simulation

To explore the effect of sequence dissimilarity, we selected the genome of the human-associated bacteria *Streptococcus mutans* UA159 (Ajdić *et al.*, 2002) as our model organism. Using the tool msbar from the EMBOSS package (v6.5.7.0; https://github.com/pjotrp/EMBOSS), we mutated the original reference sequence with a combination of substitutions, duplications, insertions and deletions to create a total of five target genomes with different divergence levels. Changes were introduced randomly as point (1), codon (3), or blocks with minimum size 1 and maximum size 10. The first target genome, denominated “None”, served as a control and contained zero genomic changes. For the two “low divergence” genomes, we allowed the introduction of ~0.0006% (1,120/2,030,936) and 0.002%

(3,790/2,030,936) genomic changes respectively, equivalent to the range of expected changes accumulated in 10,000 years, calculated by multiplying the mutation rate (0.112–0.379 substitutions/genome/year) estimated experimentally for bacteria (Cornejo *et al.*, 2013; Ochman, 2003) [Scott *et al.*, 2008]. For the two "high divergence" genomes we allowed the introduction of 0.3% (6,092) and 3%(60,928) (see Figure 01).

Contamination context and damage profile simulation

To explore the effect of non-endogenous sequences on the reconstruction of whole genomic sequence we prepared three datasets with different contamination profiles and used gargammel (Renaud *et al.*, 2017) (https://github.com/grenaud/gargammel) to generate simulated Illumina pair end reads with aDNA damage profiles (i.e. cytosine deamination and short sequence lengths).

- Dataset 1: Reconstruction of Genomes without contamination

  To establish the baseline efficiency of the methods in a contamination-free scenario, we used the five target genomes with different divergence levels and simulated on each three levels of cytosine deamination (i.e. 0.1, 0.3, 0.5), for a total of 15 subsets. Each subset contained only endogenous pair-end sequences of the target genomes at a depth of coverage of 5X.

- Dataset 2: Reconstruction of genomes with contamination (Metagenomic | interspecific)

  To simulate background microbial contamination from a metagenomic context and to evaluate the effect of non-endogenous reads on the reconstruction of bacterial genomes, we generated simulated Illumina pair-end reads with aDNA damage profiles of each target genome plus 47 bacterial genomes (28 of those found in the Human Oral Microbiome) from 11 different Phyla (Table 03). Each subset contained simulated reads of the target genomes at a 5X depth of coverage equivalent to a 10% of the total reads, and 90% non-endogenous reads.

- Dataset 3: Reconstruction of genomes with contamination of similar species

  We generated a dataset with 13 bacterial genomes from the genus Streptococcus (Table 04) to evaluate the ability of these approaches to reconstruct the original target genomes when intraspecific contamination is present. Each dataset contained simulated sequencing data with aDNA damage profiles and a composition of 80% non-endogenous reads and 20% endogenous reads, equivalent to a 5X depth of coverage of the target genomes.

Finally, we used AdapterRemoval (v2.2.0, https://github.com/MikkelSchubert/adapterremoval) to collapse reads and remove adapters, allowing a minimal length of 25, minimal quality of 4, and minimum alignment length of 11.

Genome Reconstruction Approaches

We tested three different approaches for whole genome reconstruction: reference mapping assembly; reference guided de novo assembly; and iterative mapping. For the mapping assembly approach we used the programs BWA and Bowtie2, each with different parameters combinations used in published palaeomicrobiology studies (Table 01 & 02), and the original genomic sequence of Streptococcus mutans UA159 as the reference sequence. For the reference assisted de novo assembly approach we used VelvetOptimiser with kmer sizes 21-45 to create contigs de novo, and used the same algorithms of the reference mapping approach to map the resulting contigs to the reference sequence of S. mutans UA159. Finally for the iterative mapping approach we used MIRA 4 to create an initial mapping assembly to the reference sequence and MITObim (v1.9) to iteratively map reads, and MITObim (v1.9) with denovo assembly parameters.

- we changed the number of mismatches allowed by MITObim to check the stringency...

Evaluation and Comparison Stats

To evaluate the performance of the different approaches for whole genome assembly we used three main parameters: Percentage of original sequence recovered, Fidelity of the recovered sequence to the original genome, and the ratio of non-endogenous reads vs. endogenous reads.

For the reference mapping and reference guided de novo assembly, we used samtools to filter reads with mapping quality <=30 and to calculate basic mapping statistics (Coverage, Depth of Coverage, Number of sequences mapped, etc.). We removed duplicates using picard tools and generated a consensus sequence fasta file using bcftools. We split the consensus sequence in "contigs" whenever we found more than 5 "N". We used these "contigs" to evaluate the assembly using the Quality ASsesment Tool for genome assemblies (QUAST) and the original target genomes.

RESULTS

In this study, we explore how different genome reconstruction approaches are affected by characteristics commonly found in ancient metagenomic datasets. We used the genomic reference sequence of Streptococcus mutans UA159 and simulated insertions, deletions, and substitutions on it at five increasing levels to generate five target genomes with variable divergence levels. We include each target genome in three different contexts of contamination (No-Contamination (0C100E); inter-species Contamination (90C10E); Intra-genera Contamination (80Smu20E)), and simulated reads with ancient damage profiles with three increasing levels of deamination (0.1, 0.3, 0.5), for a total of 45 read pools. Then, we used reference mapping, reference assisted de novo assembly, and iterative mapping approaches to reconstruct the original target genomes from each pool.

In the reference mapping approach, we used the original reference sequence of *S. mutans* UA159 and mapped reads with the programs *Bowtie2* and *BWA* using three different sets of parameters (including parameters optimized for aDNA), and *MIRA* with parameters specific for Illumina reads. In the reference assisted assembly, we generated contigs *de novo* using Velvet (v.##) through VelvetOptimiser and mapped the resulting contigs to the reference with the previously mentioned mappers. Finally, in the iterative mapping approach, we use MITObim with mapping and *de novo* initial steps and allowing different mismatch values. The resulting bam alignments were filtered by Mapping Quality (MQ>=30), and consensus files were called using samtools(v.##)/bcftools (v.##). We then divide the final consensus sequence of each approach and divide them in "consensus contigs" wherever we found more than 5 N's. Then, using QUAST (QUalitative Assessment tool), we align the resulting consensus contigs to the original target genomes and evaluate the reconstruction. QUAST allow us to analyze and compare the different assemblies, and we use it to compare the results of the different approaches.

As a measurement of efficiency, we calculated the number of endogenous reads that were recovered for each approach. Figure # shows the percentage of endogenous reads mapped with high confidence from the total endogenous reads available in each pool. This value was calculated by counting the number of reads from the target genomes that were mapped with mapping quality 30 or more and dividing this count by the total number of the simulated reads generated from that target genome. This measure allows us to compare how much 'real' information could be recovered with the different methods and tools at all levels of divergence and deamination. The amount of endogenous information that the mapping approaches were able

Result Figures

Introductory paragraph

Effects of Divergence

Effects of Contamination

Effects of Damage profiles

Variables

```
##  [1] "Divergence"                 "Contamination"
##  [3] "Deamination"                "TypeReads"
##  [5] "Method"                     "Tool"
##  [7] "TotalReadPairs"             "TotalCollapsedReads"
##  [9] "TotalCollapsedEndogenous"   "TotalContigs"
## [11] "TotalReadsSAM"              "MappedReads"
## [13] "PercMappedReads"            "MQ30MappedReads"
## [15] "MQ30Endogenous"             "MQ30Coverage"
## [17] "MQ30DepthCov"               "MQ30EndoWithinStrict"
## [19] "MQ30EndoOutsideStrict"      "MQ30EndoSumStrict"
## [21] "MQ30EndoWithin50"           "MQ30EndoOutside50"
## [23] "MQ30EndoSum50"              "EndoRNA"
## [25] "TotalRNA"                   "TotalEndogenous"
## [27] "TotalEndogenousMasked"
```

```
##  [1] "Divergence"
##  [2] "Contamination"
##  [3] "Deamination"
##  [4] "TyperReads"
##  [5] "Method"
##  [6] "Tool"
##  [7] "QUASTfilter"
##  [8] "Num.contigs..GreaterOrEqual.0.bp."
##  [9] "Num.contigs..GreaterOrEqual.1000.bp."
## [10] "Num.contigs..GreaterOrEqual.5000.bp."
## [11] "Num.contigs..GreaterOrEqual.10000.bp."
## [12] "Num.contigs..GreaterOrEqual.25000.bp."
## [13] "Num.contigs..GreaterOrEqual.50000.bp."
## [14] "Total.length..GreaterOrEqual.0.bp."
## [15] "Total.length..GreaterOrEqual.1000.bp."
## [16] "Total.length..GreaterOrEqual.5000.bp."
## [17] "Total.length..GreaterOrEqual.10000.bp."
## [18] "Total.length..GreaterOrEqual.25000.bp."
## [19] "Total.length..GreaterOrEqual.50000.bp."
## [20] "Num.contigs"
## [21] "Largest.contig"
## [22] "Total.length"
## [23] "Reference.length"
## [24] "GC..Percent."
## [25] "Reference.GC..Percent."
## [26] "N50"
## [27] "NG50"
## [28] "N75"
## [29] "NG75"
## [30] "L50"
## [31] "LG50"
## [32] "L75"
## [33] "LG75"
## [34] "Num.misassemblies"
## [35] "Num.misassembled.contigs"
## [36] "Misassembled.contigs.length"
## [37] "Num.local.misassemblies"
## [38] "Num.unaligned.mis..contigs"
## [39] "Num.unaligned.contigs"
## [40] "Unaligned.length"
## [41] "Genome.fraction..Percent."
## [42] "Duplication.ratio"
## [43] "Num.N.s.per.100.kbp"
## [44] "Num.mismatches.per.100.kbp"
## [45] "Num.indels.per.100.kbp"
## [46] "Largest.alignment"
## [47] "Total.aligned.length"
## [48] "NA50"
## [49] "NGA50"
## [50] "NA75"
## [51] "NGA75"
## [52] "LA50"
## [53] "LGA50"
## [54] "LA75"
## [55] "LGA75"
## [56] "Num.misassemblies.1"
## [57] "Num.relocations"
## [58] "Num.translocations"
## [59] "Num.inversions"
## [60] "Num.misassembled.contigs.1"
## [61] "Misassembled.contigs.length.1"
## [62] "Num.local.misassemblies.1"
## [63] "Num.unaligned.mis..contigs.1"
## [64] "Num.mismatches"
## [65] "Num.indels"
## [66] "Num.indels..LowerOrEqual.5.bp."
## [67] "Num.indels..Greater.5.bp."
## [68] "Indels.length"
```

Summary of variables

Read Stats Summaries

In this section, we show figures based on statistics calculated on reads. We compare all the variables across the different genomic reconstruction approaches; each one applied at sequence pools with a specific level of contamination, divergence, and deamination. Then, we evaluate the results and explain how each approach is affected by particular conditions.

TOTAL AVAILABLE READS PER POOL

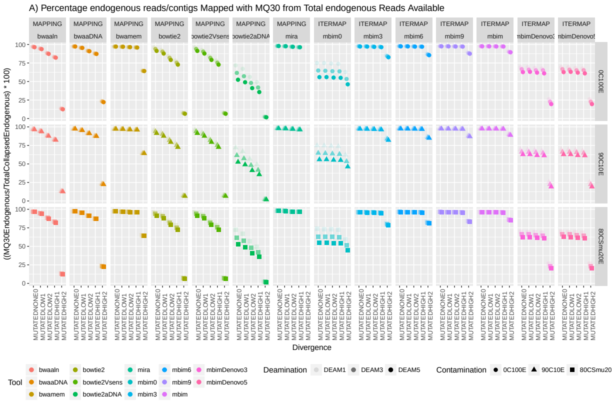
We plotted the initial amount of sequencing reads across the different methods and tools utilized.

- **Dataset 1 (0C100E)** contains only reads of the target genomes with a 5X depth of coverage (approx. 150,000 reads each). This dataset allows us to study the impact of aDNA damage profiles in a "contamination-free" scenario.
- **Dataset 2 (90C10E)** contains a combination of bacteria from different phyla in addition to the target genome reads with a ratio of 9:1 for a total of 1.5 million reads. This dataset simulates the influence of metagenomic data, and allow us to study how inter-specific contamination can affect the reconstruction of ancient microbial genomes.
- **Dataset 3 (80CSmu20E)** contains reads from bacteria from the same genera (Streptococcus) in addition to the target genome reads with a ratio of 4:1. This dataset simulates the presence of highly similar bacteria and allows us to study how they affect the reconstruction of ancient microbial genomes with high levels of heterogeneity.

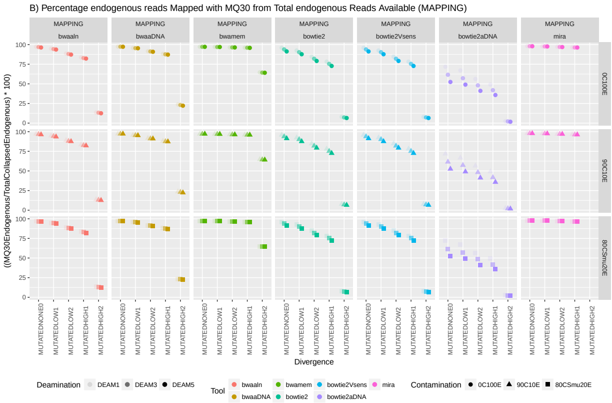
PERCENTAGE ENDOGENOUS READS MAPPED WITH MQ>=30 FROM TOTAL ENDOGENOUS READS AVAILABLE

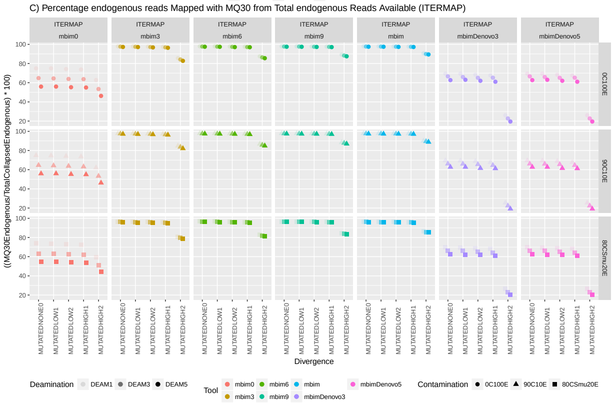
The following graphs show the percentage of endogenous reads mapped with high confidence from the total endogenous reads available in each pool. This value was calculated by counting the number of reads from the target genomes that were mapped with mapping quality 30 or more and dividing this count by the total number of the simulated reads generated from that target genome. This measure allows us to compare how much 'real' information could be recovered with the different methods and tools at all levels of divergence and deamination. The amount of endogenous information that the mapping approaches were able to recover is significantly affected by the divergence between the reference genome used, and the target genome reads. Notably, *MIRA* allows the recovery of most of the endogenous reads available and is the mapping approach that is less affected by divergence. From the other tools, *BWA mem* is the second best approach, allowing the recovery of the highest number of reads on all the divergence levels, and recovering almost three times more endogenous reads than the next approach (*BWA aDNA*) in the highest level of divergence. The approaches based on *Bowtie2* recovered the lowest amount of endogenous reads, with *Bowtie2 aDNA* being the worst performer. Notably, the deamination level does not appear to have a significant effect on the number of reads that the approaches based on *BWA* were able to map, but it does affect the *Bowtie2* approaches, with *Bowtie2 aDNA* being the most affected by increases in cytosine deamination. Among the iterative approaches, the ones with an initial mapping step were able to recover a more considerable amount of endogenous reads, even at the highest level of divergence, and the levels of deamination did not seem to have a significant impact on them. However, as expected, the most restrictive iterative approach, which did not allow for mismatches (*MITObim0*), was the worst performer and was greatly affected by increasing levels of deamination. On the other hand, the ones including a *de novo* assembly step were able to recover only about 60% of all the endogenous reads available in the first four levels of divergence (similar to the approach *MITObim0*) and less than 25% in the highest level of divergence.

```
## Warning: Removed 9 rows containing missing values (geom_point).
```



```
## Warning: Removed 9 rows containing missing values (geom
```



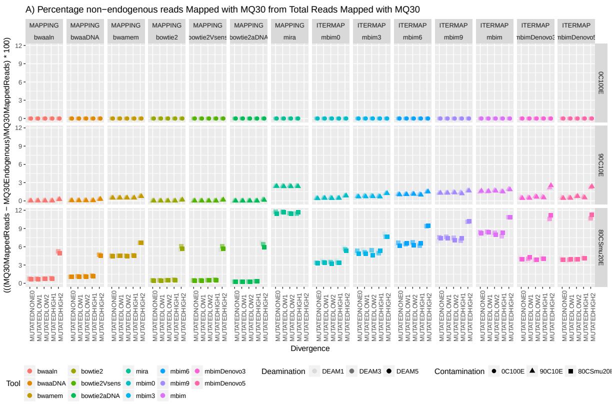


PERCENTAGE OF NON-ENDOGENOUS READS MAPPED WITH MQ>=30 FROM TOTAL READS MAPPED WITH MQ30

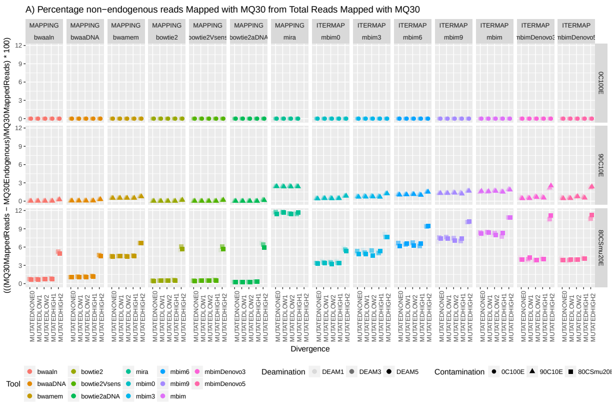
The following graphs show the amount of non-endogenous reads mapped with high confidence (MQ>=30) from the total amount of mapped reads with MQ>=30. This value is a measure of the accuracy of the tools and gives us an idea of how much 'contamination' could be added from other genomes even after filtering by mapping quality. In this case, we can observe that the type of contamination has a substantial impact on the number of non-endogenous reads recovered, with the dataset composed of closely related species being the most affected across all tools and methods. The iterative approach is the most affected, allowing the introduction of more than 2% of non-endogenous reads in the "metagenomic" dataset, and more than 10% when highly related species are present, and the divergence between the reference and the target genome is the highest. From the mapping approach, *MIRA* is the most affected mapper, followed by *BWA mem*. The approaches using *Bowtie2* performed the best and only included around 6%

non-endogenous reads when highly similar reads were present, and the divergence was the highest. Again, the amount of deamination in the reads was not significant.

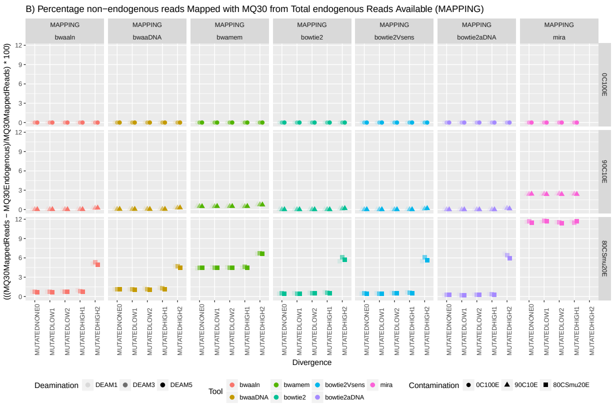
```
## Warning: Removed 9 rows containing missing
```

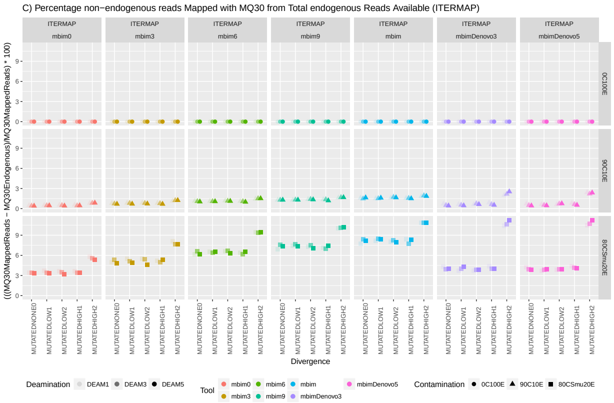


```
## Warning: Removed 9 rows containing missing values (geom_point).
```



```
## Warning: Removed 9 rows containing missing
```



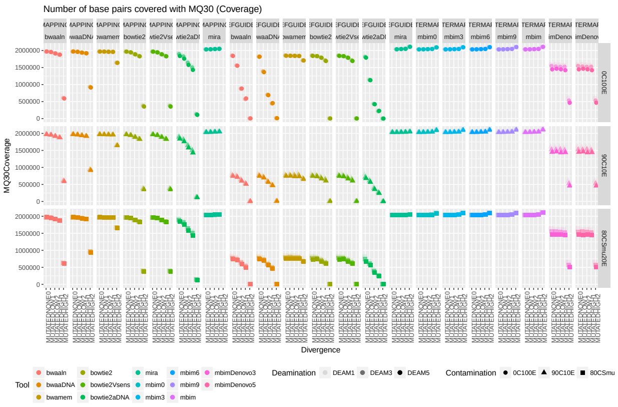


COVERAGE, GENOME FRACTION PERCENT, TOTAL ALIGNED LENGTH

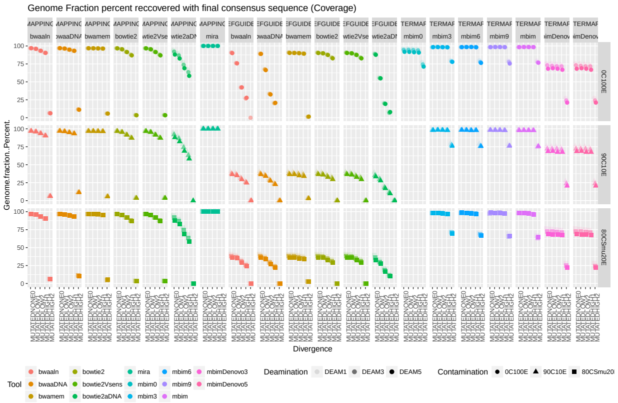
Number of base pairs covered with mapping quality MQ>=30 accross three methods and the different tools used on each.

- Coverage is calculated by counting every position with depth of coverage equal o greater than 1.
- Genome Fraction percent is the percentage of aligned bases in the target genome covered by aligning the resulting “contigs” of the consensus sequence generated for each alignment.
- Total aligned lenght is the toal number of aligned bases in the assembly

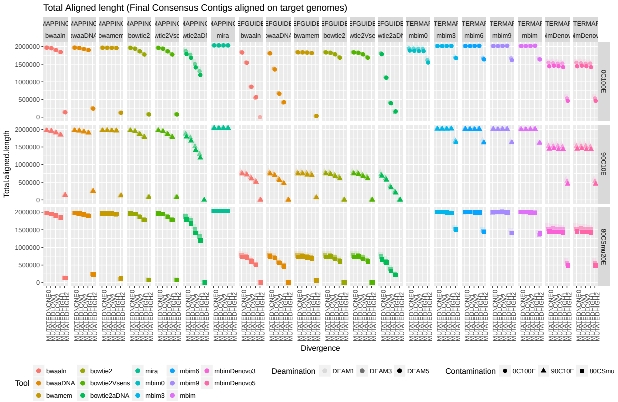
```
## Warning: Removed 12 rows containing missing values (ge
```



```
## Warning: Removed 11 rows containing missing values (ge
```



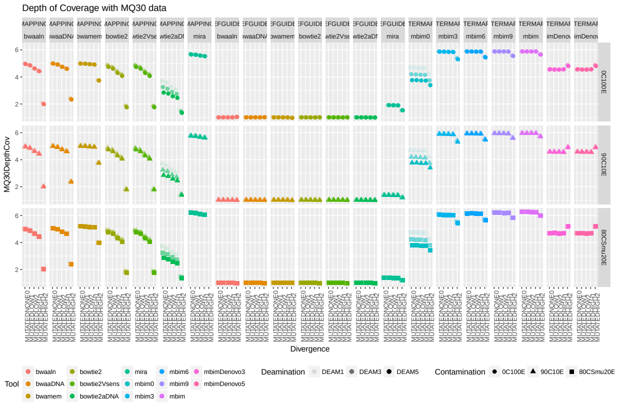
```
## Warning: Removed 11 rows containing missing values (geom_point).
```



DEPTH OF COVERAGE

Depth of coverage is calculated with samtools, and is the average of the depth of coverage across the full length of the genome.

```
## Warning: Removed 12 rows containing missing
```



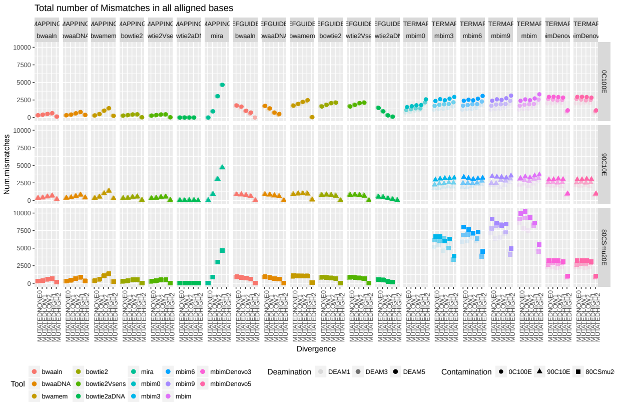
QUAST Statistics

NUMBER OF MISMATCHES

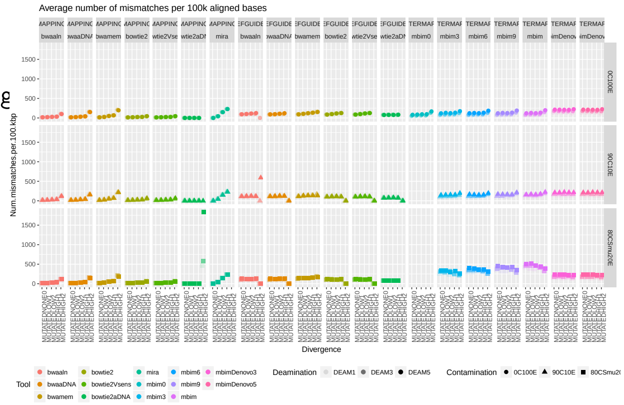
Using QUAST we aligned the resulting consensus contigs of each approach and evaluated the reconstruction.

- **Mismatches**: Number of mismatches in all aligned bases.
- **Mismatches per 100kbp**:Average number of indels per 100,000 alinged bases. True SNPs and sequencing errors are not distinguished and are counted equally.

```
## Warning: Removed 11 rows containing missing values (geo
```

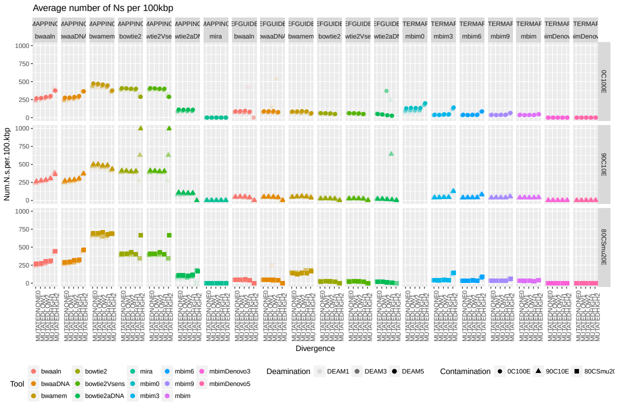


```
## Warning: Removed 11 rows containing missing values (geo
```



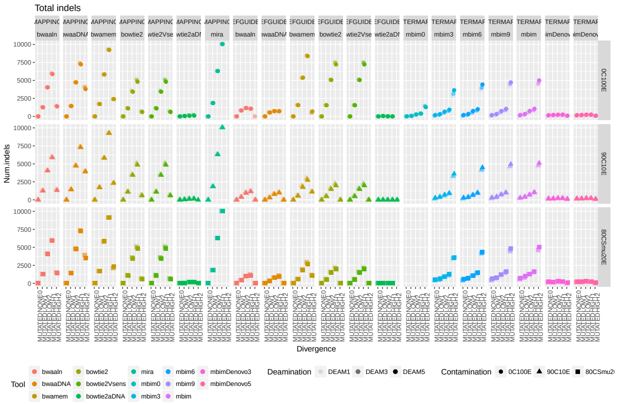
N's per 100k bases aligned

Total number of uncalled bases (N's) in the assembly

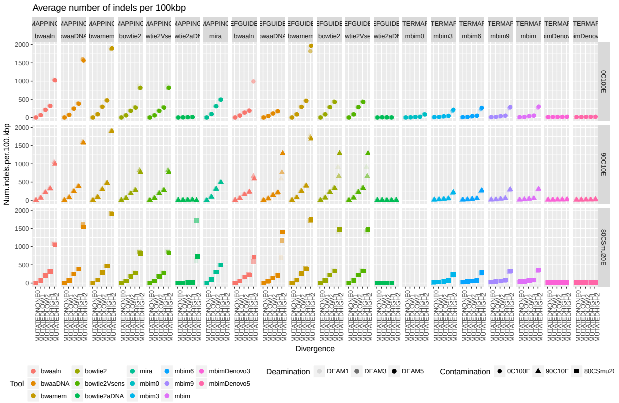


INDELS

```
## Warning: Removed 11 rows containing missing
```



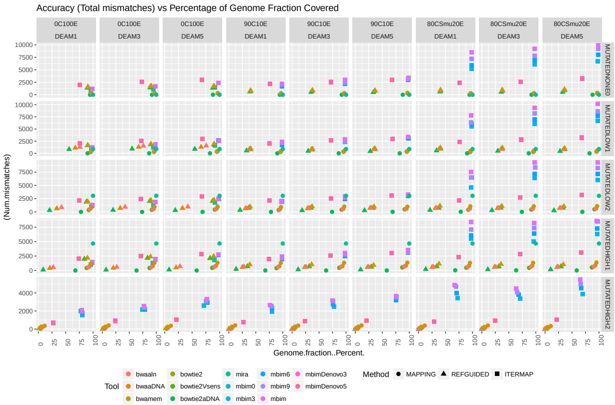
```
## Warning: Removed 11 rows containing missing values (geom_point).
```



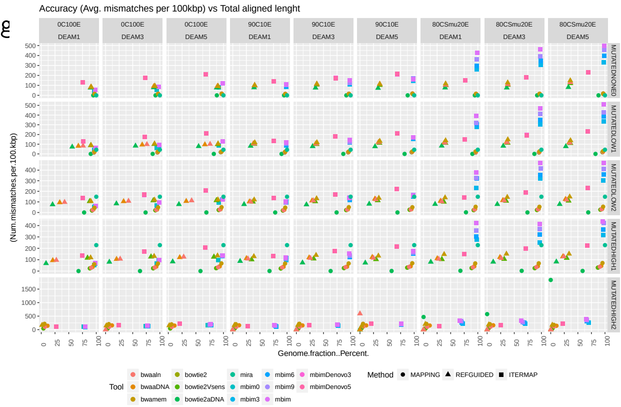
COMBINATIONS

GENOME FRACTION PERCENTAGE vs MISMATCHES (Accuracy)

```
## Warning: Removed 11 rows containing missing values (geom_point).
```

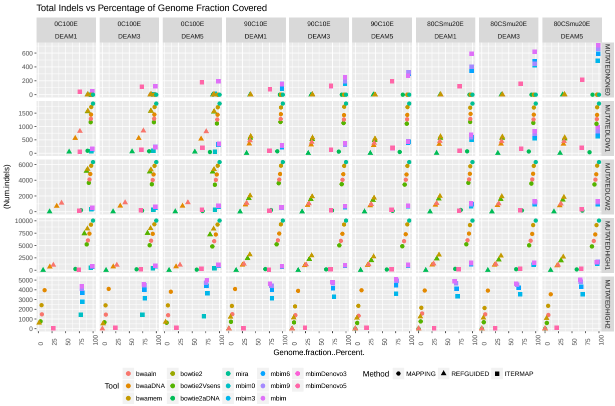


```
## Warning: Removed 11 rows containing missing values (geo
```

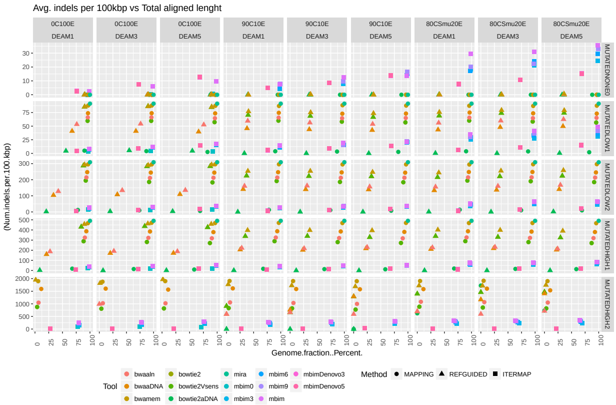


GENOME FRACTION PERCENTAGE vs INDELS

```
## Warning: Removed 11 rows containing missing values (geo
```

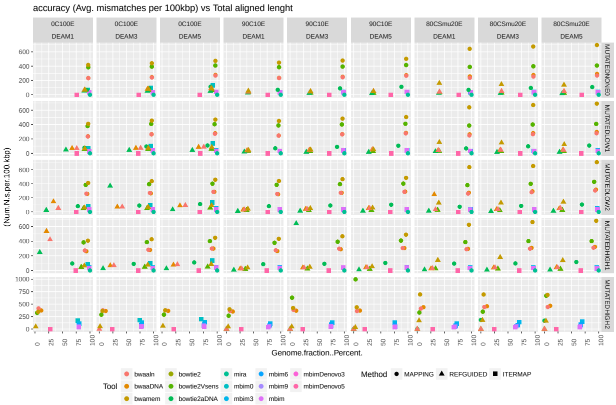


```
## Warning: Removed 11 rows containing missing values (geo
```



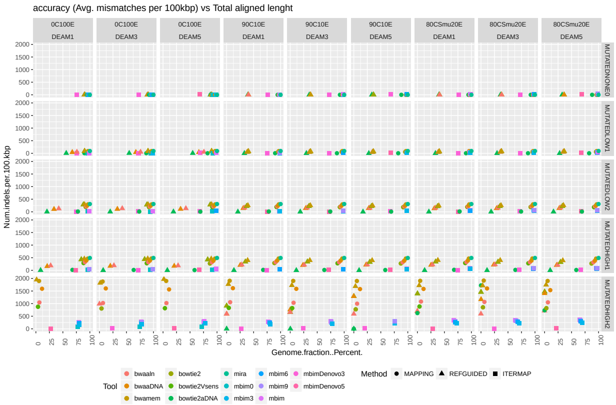
GENOME FRACTION PERCENTAGE vs N's

```
## Warning: Removed 11 rows containing missing values (geom_point).
```



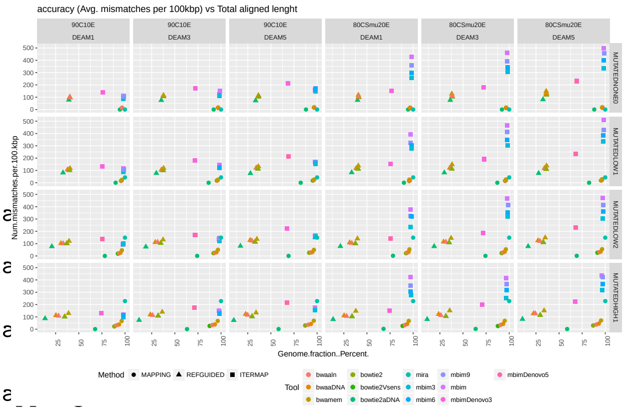
```
## Warning: position_dodge requires non-overlapping x intervals

## Warning: position_dodge requires non-overlapping x intervals

## Warning: position_dodge requires non-overlapping x intervals

## Warning: position_dodge requires non-overlapping x intervals

## Warning: position_dodge requires non-overlapping x intervals

## Warning: position_dodge requires non-overlapping x intervals

## Warning: position_dodge requires non-overlapping x intervals

## Warning: position_dodge requires non-overlapping x intervals

## Warning: position_dodge requires non-overlapping x intervals

## Warning: position_dodge requires non-overlapping x intervals

## Warning: position_dodge requires non-overlapping x intervals

## Warning: position_dodge requires non-overlapping x intervals
```

```
## Warning: position_dodge requires non-overlapping x inte

## Warning: position_dodge requires non-overlapping x inte

## Warning: position_dodge requires non-overlapping x inte

## Warning: position_dodge requires non-overlapping x inte

## Warning: position_dodge requires non-overlapping x inte

## Warning: position_dodge requires non-overlapping x inte

## Warning: position_dodge requires non-overlapping x inte

## Warning: position_dodge requires non-overlapping x inte

## Warning: position_dodge requires non-overlapping x inte

## Warning: position_dodge requires non-overlapping x inte

## Warning: position_dodge requires non-overlapping x inte

## Warning: position_dodge requires non-overlapping x inte

## Warning: position_dodge requires non-overlapping x inte

## Warning: position_dodge requires non-overlapping x inte

## Warning: position_dodge requires non-overlapping x inte

## Warning: position_dodge requires non-overlapping x inte

## Warning: position_dodge requires non-overlapping x inte

## Warning: position_dodge requires non-overlapping x inte

## Warning: position_dodge requires non-overlapping x inte

## Warning: position_dodge requires non-overlapping x inte

## Warning: position_dodge requires non-overlapping x inte

## Warning: position_dodge requires non-overlapping x inte

## Warning: position_dodge requires non-overlapping x inte

## Warning: position_dodge requires non-overlapping x inte

## Warning: position_dodge requires non-overlapping x inte

## Warning: position_dodge requires non-overlapping x inte

## Warning: position_dodge requires non-overlapping x inte
## Warning: position_dodge requires non-overlapping x intervals

## Warning: position_dodge requires non-overlapping x intervals

## Warning: position_dodge requires non-overlapping x intervals

## Warning: position_dodge requires non-overlapping x intervals

## Warning: position_dodge requires non-overlapping x intervals

## Warning: position_dodge requires non-overlapping x intervals

## Warning: Removed 11 rows containing missing values (geom_point).
```

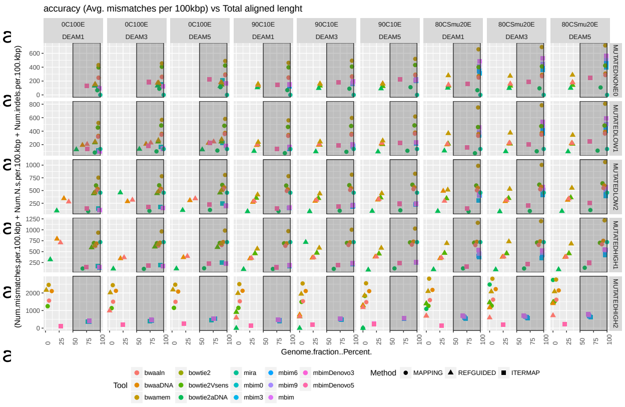


```
## Warning: position_dodge requires non-overla

## Warning: position_dodge requires non-overla

## Warning: position_dodge requires non-overla

## Warning: position_dodge requires non-overla

## Warning: position_dodge requires non-overlapping x intervals

## Warning: position_dodge requires non-overlapping x intervals

## Warning: position_dodge requires non-overla

## Warning: position_dodge requires non-overla

## Warning: position_dodge requires non-overla

## Warning: position_dodge requires non-overla

## Warning: position_dodge requires non-overla

## Warning: position_dodge requires non-overla

## Warning: position_dodge requires non-overlapping x intervals

## Warning: position_dodge requires non-overlapping x intervals
```

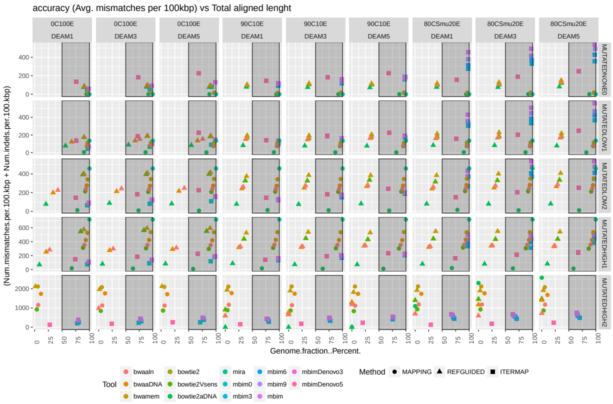
```
## Warning: position_dodge requires non-overlapping x inte

## Warning: position_dodge requires non-overlapping x inte

## Warning: position_dodge requires non-overlapping x inte

## Warning: position_dodge requires non-overlapping x inte

## Warning: position_dodge requires non-overlapping x inte

## Warning: position_dodge requires non-overlapping x inte

## Warning: position_dodge requires non-overlapping x inte

## Warning: position_dodge requires non-overlapping x inte

## Warning: position_dodge requires non-overlapping x inte

## Warning: position_dodge requires non-overlapping x inte
```

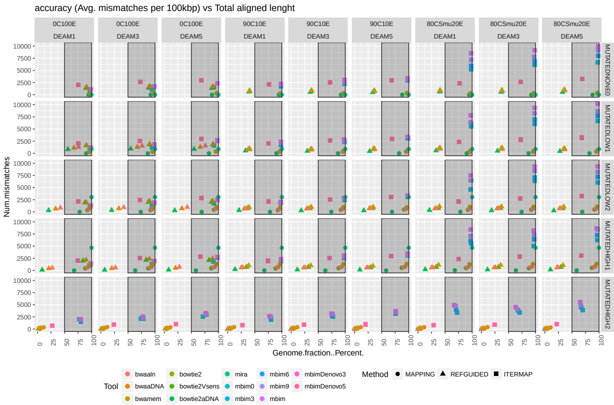


```
## Warning: Removed 11 rows containing missing values (geo
```

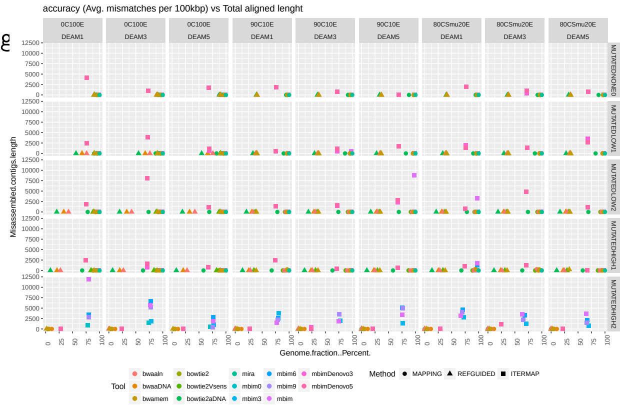


```
## Warning: Removed 11 rows containing missing values (geo
```

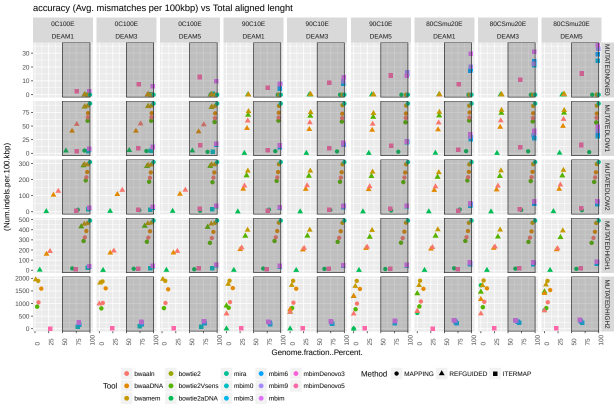




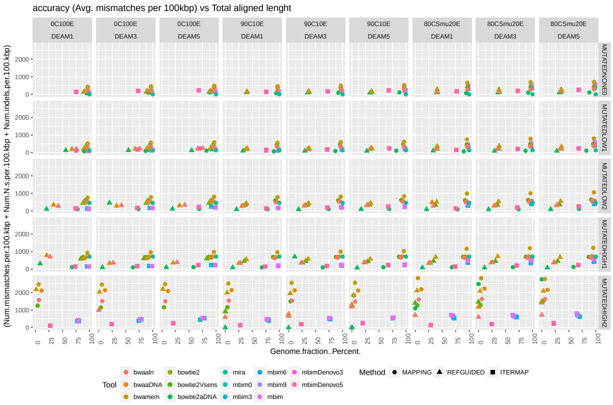
## Warning: Removed 11 rows containing missing values (geo



## Warning: Removed 11 rows containing missing



## Warning: Removed 11 rows containing missing values (geom_point).



## Warning: Removed 11 rows containing missing values (geom_point).