Supplementary Material of:

Circlator: automated circularization of genome assemblies using long sequencing reads

Martin Hunt [1], Nishadi De Silva [1], Thomas D Otto [1], Julian Parkhill [1], Jacqueline A Keane [1] and Simon R Harris [1]

[1]Wellcome Trust Sanger Institute, Wellcome Trust Genome Campus, Cambridge, CB10 1SA, UK

1 Reference dnaA genes

The panel of reference dnaA genes was produced as follows. The term:
`'dnaA[sym] AND ("srcdb refseq"[Properties] AND alive[property])'`
was used to search the 'gene' database at the NCBI website `http://www.ncbi.nlm.nih.gov/`. All the results were saved as a 'Tabular (text)' file, which was used as input to the script `get_dnaA.pl`. This script downloads the sequences, and retains only those that are in the length range 1000-1600bp, begin with a start codon and end with a stop codon.

2 PacBio circularization protocol

The following is a verbatim copy of the PacBio circaulariztion protocol, obtained from `https://github.com/PacificBiosciences/Bioinformatics-Training/wiki/Circularizing-and-trimming` on 22nd June 2015.

1. Open up the `polished_assembly.fasta` file in a text editor.
2. In the middle of the circular contig introduce a break, i.e. a new line '`>Break`'. It does not matter where in the sequence you introduce the break, but the sequence immediately after the break will be the start of your circularized sequence.
3. `toAmos -s polished_assembly.fasta -o circularized.afg`
4. `minimus2 circularized`
5. Minimus will overlap and join the ends of the two contigs, the resulting `circularized.fasta` file should contain one contig for the sequence in which you introduced a break.
6. Any sequence that was not circularized, or extra contigs in which you did not introduce a break will be in `circularized.singletons.seq`, and can be added back to the `circularized.fasta` file
   `cat circularized.singletons.seq >> circularized.fasta`
7. The sequence that was overlapped now needs to be quiver corrected, to do this simply import the circularized.fasta file into SMRT Portal as a reference and run a resequencing job with the raw data. Problems with the coverage in the region which was overlapped could indicate an issue with the circularization and the possibility that the molecule was not circular, or was split at a repeat that has not been accounted for.

3 *Plasmodium falciparum* apicoplast

To improve the *P. falciparum* 3D7 reference sequence we sequenced the genome with Illumina short insert (accessions ERR007655-6) and 454 8kb insert libraries (accessions ERR102953-4). Read coverage analysis of the Illumina data showed that the last 5kb of the apicoplast was duplicated. The duplication contained fewer than

five heterozygous SNPs, indicating that the two copies are nearly identical. With the help of the 454 large fragment library, we manually arranged the two copies as an inverted repeat separated with 16 unique bases. Further we confirmed that the plastid is circular, as expected. Previous attempts to circularize the apicoplast and confirm the orientation of the unique sequence by PCR were unsuccessful.

We used the 454 reads to confirm the overall structure of the new, circularized, sequence made by Circlator. The reads were mapped to the complete HGAP assembly using SMALT (`https://www.sanger.ac.uk/resources/software/smalt/`) version 0.7.0.1 with the options `-x -r 1 -i 13000 -y 0.8` and reads, plus their mates, that mapped to the HGAP contig corresponding to the apicoplast were retained. These reads were mapped independently to the manually created reference sequence and to the output of Circlator using SMALT with the same settings as above. Both sequences were mapped to because they have different start positions, allowing the complete circular sequence to be verified. See Supplementary Figure S4, showing the reads mapped to both sequences, confirming the boundaries of the inverted repeat sequence and the point at which Circlator rearranged the start point of the sequence.

4 Comparing assemblies and reference genomes

Supplementary Figures S5–S21 show the input assembly compared to the reference sequence, and the reference sequence compared to the output of the BLAST-based method, Circlator, and Minimus2. These figures were generated using the supplementary script `act_cartoon.py`. In each figure, the top genome is the input assembly, the middle genome is the reference, and the bottom assembly is the output of one of the three circularization methods.

Nucmer matches are shown between the genomes in blue and pink. Hits on the same strand are shown in blue, and those on opposite strands are coloured pink. All matches are shown that have a minimum identity of 98% and are either at least 5000bp long or at least 3% of the length of the input or output assembly contig length. The only exception is the nanopore data in Supplementary Figure S20, where the minimum identity was reduced to 85%.

The input assembly contigs that were merged are coloured yellow (this only applies to Circlator and the Minimus2-based method). Input assembly contigs that were removed are coloured red (this only applies to Circlator). Any output assembly contig flagged as circular by the tool that made it is coloured green (applicable to all three tools).

The reference genome has three plots. The top and bottom plots (colored black) show the number of nucmer matches between the reference and the input and output assemblies. They are normalised so that they have the same y axis scale, to allow comparison. The upper plot is generated from hits to the input assembly, and the lower plot is generated from hits to the output assembly. These plots can be used to identify overlapping sequence that is removed during the circularization process. For example, on sample NCTC13616 Circlator and the BLAST-based method removed one copy of the repetitive sequence at the start and end of the input assembly contig (see Supplementary Figure S17).

The centre plot (in blue) marks repeats in the reference genome. A region is

	CPU (s)	Wall clock (s)	RAM (MB)
BLAST	2.14	7	26
Circlator	394.95	390	245
Minimus2	28.63	38	62

Table S5: Median CPU time, wall clock time and peak memory usage calculated across all datasets.

counted as a repeat if it has a nucmer hit to elsewhere in the reference genome. Hits must be at least 250bp long and have a percent identity of at least 98. These hits frequently match assembly contig ends, confirming that contigs often end with repeat sequences. See for example Supplementary Figure S12.

5 Run time and memory

The values used for peak memory usage were those reported by the compute farm job scheduling software Platform Load Sharing Facility (LSF). It polls the memory usage every minute and reports the maximum value when a job finishes. It also reports the total CPU time and wall clock time. The running times and memory usage are summarised in Supplementary Figure S24. Although Circlator has a longer running time than the other methods, the median time is under 7 minutes (Supplementary Tables S4 and S5). This is still relatively insignificant compared to the run time of an assembly (typically several hours, using multiple threads). The memory usage of all the tools is low, with a maximum memory of 490MB used by Circlator on sample NCTC13348.

The majority of the running time of Circlator is spent mapping the reads with BWA and running assemblies with SPAdes. Although we ran all analysis using one thread, the wall clock time could be reduced by using more threads with BWA and SPAdes via the option `--threads` to Circlator.

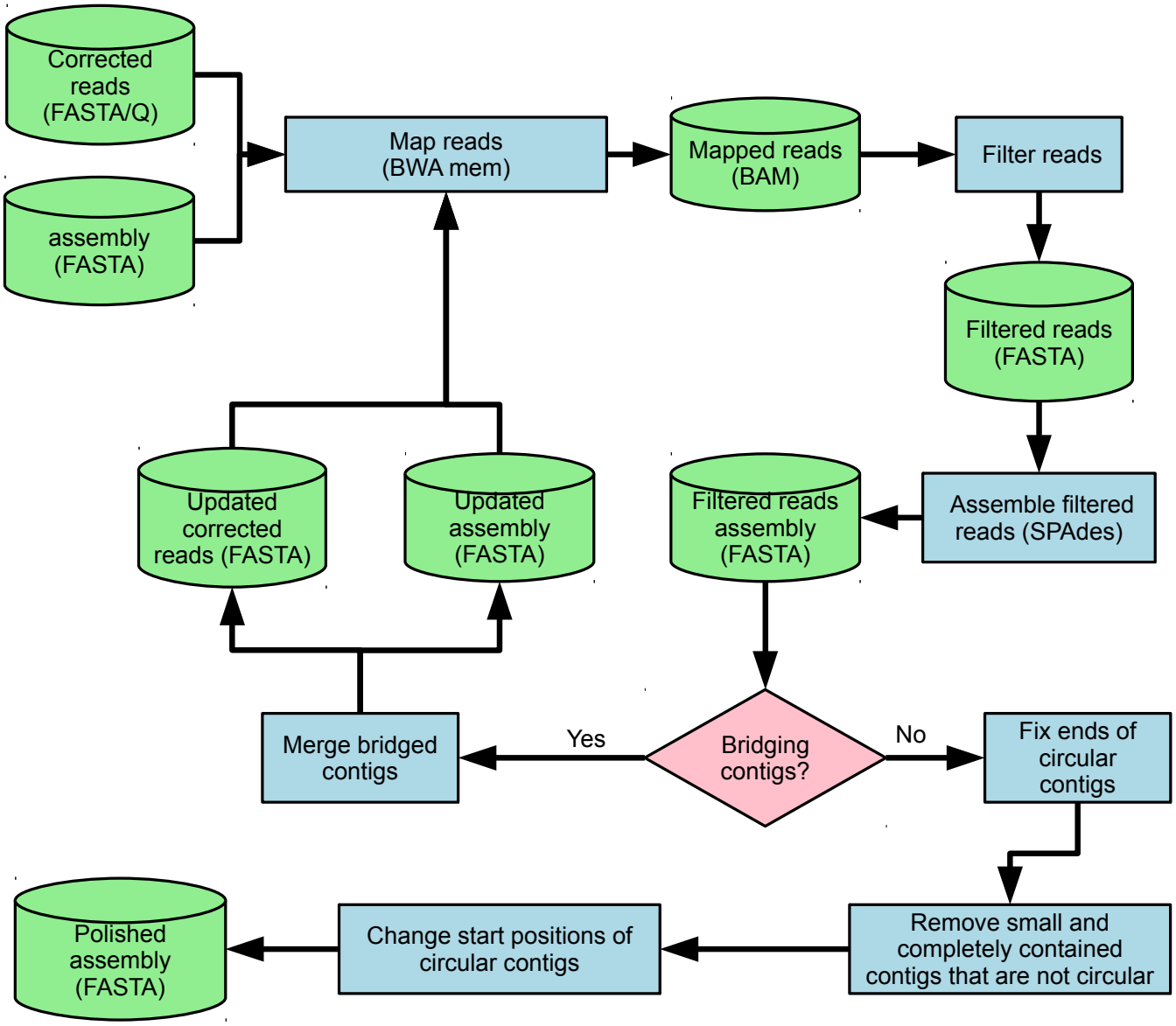

Figure S1: Workflow of Circlator

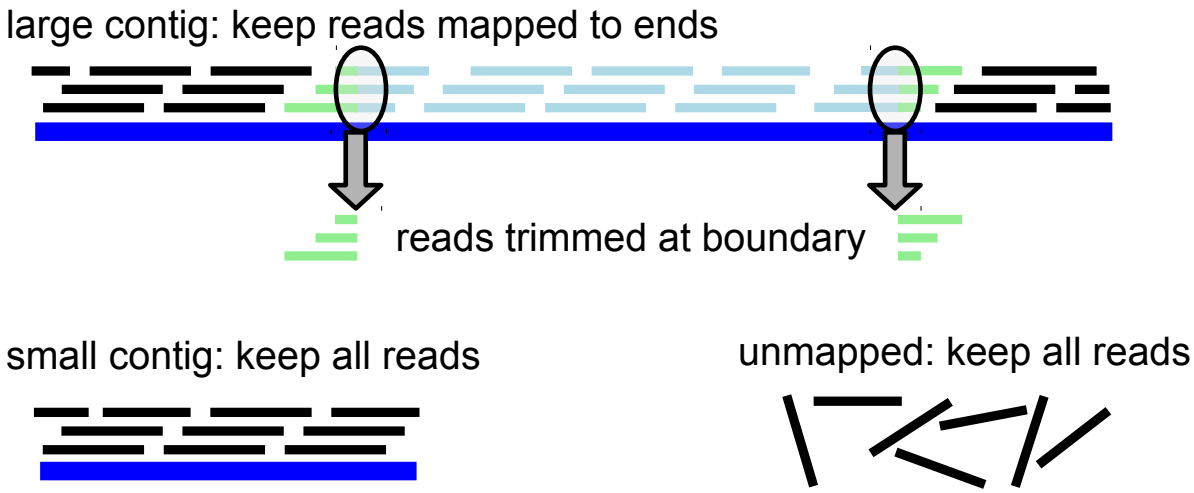

Figure S2: Read trimming and filtering used by Circlator for local assemblies

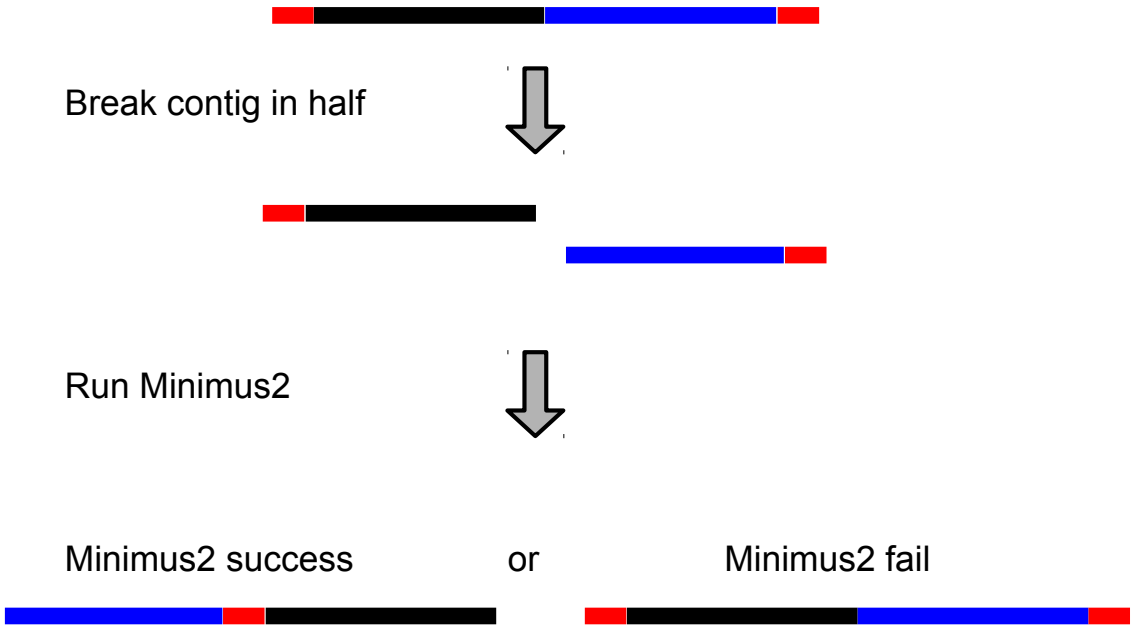

Figure S3: Using Minimus2 to circularize a single contig

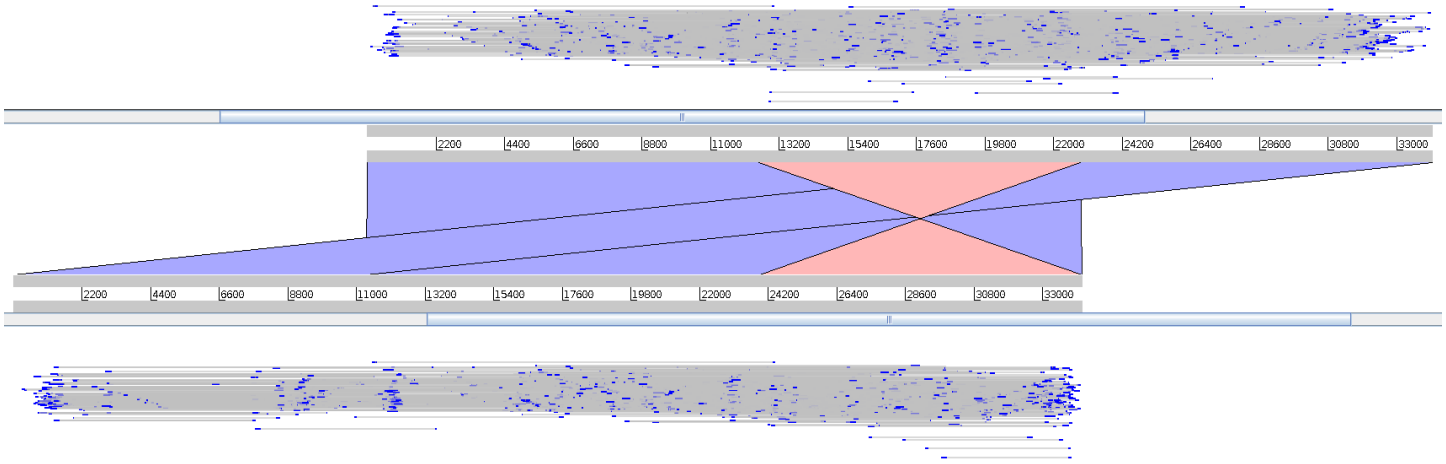

Figure S4: ACT[1] screenshot showing a comparison of the reference (bottom) *P. falciparum* apicoplast genome and the output of Circlator (top). BLAST hits between the genomes are shown in blue (hits in the same direction) and red (in the opposite direction). Proper read pairs are shown in blue, with the two read within a pair joined by a grey line ('inferred size' view). The height of a read pair is determined by the distance between the reads (using a logarithmic scale).

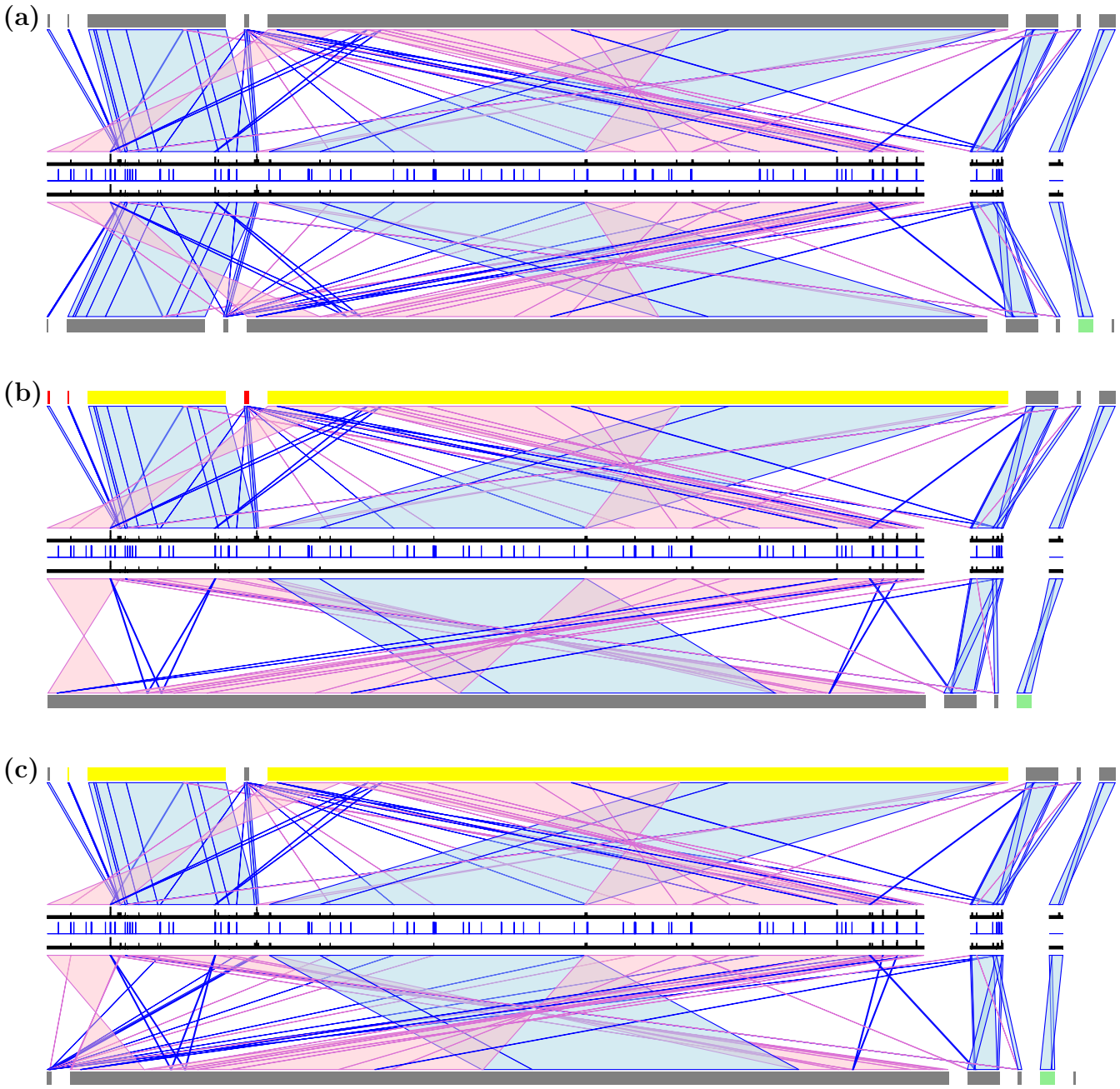

Figure S5: Comparison of circularizing NCTC sample NCTC10005 using (a) BLAST, (b) Circlator and (c) Minimus2.

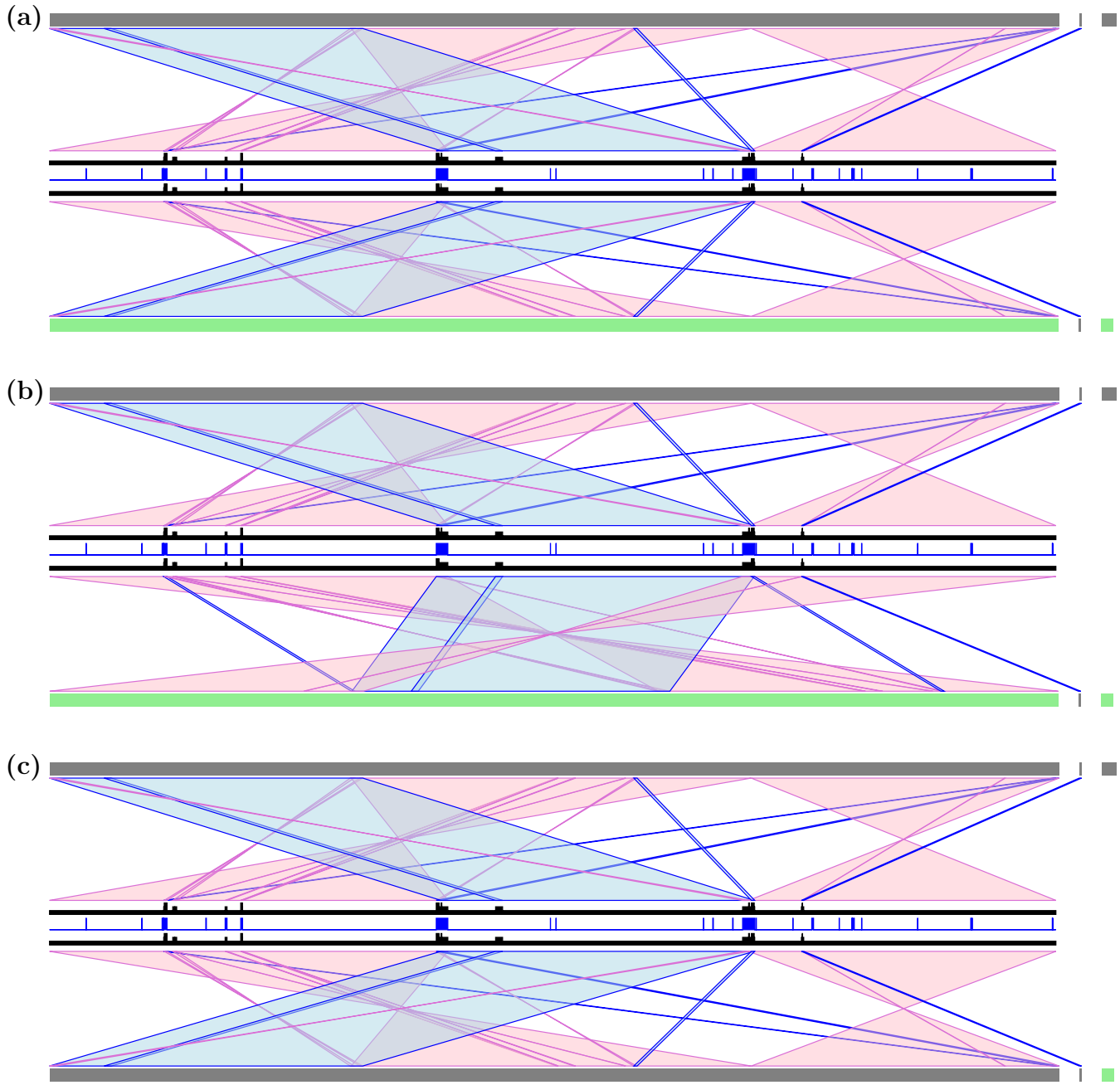

Figure S6: Comparison of circularizing NCTC sample NCTC10833 using (a) BLAST, (b) Circlator and (c) Minimus2.

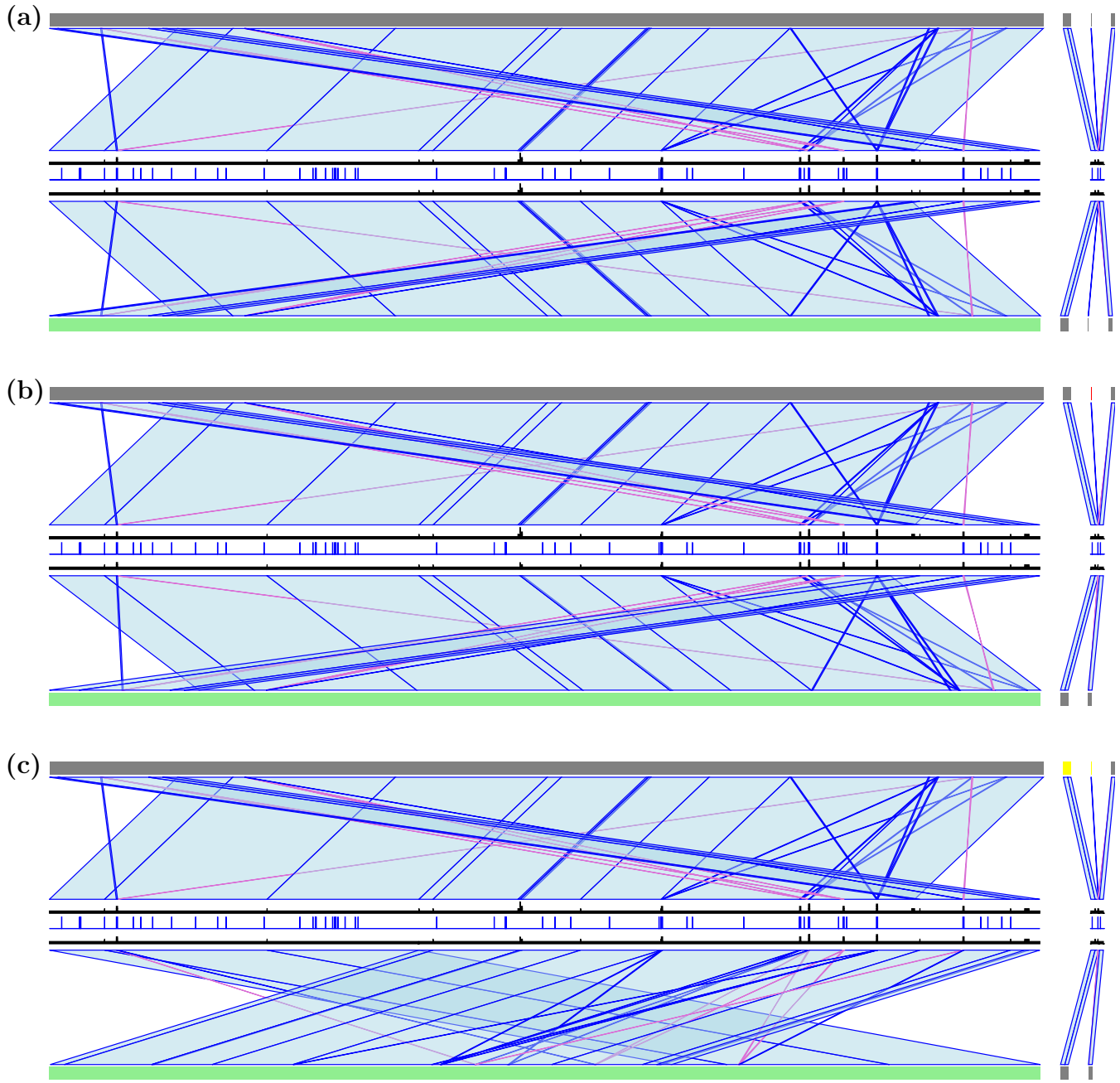

Figure S7: Comparison of circularizing NCTC sample NCTC10963 using (a) BLAST, (b) Circlator and (c) Minimus2.

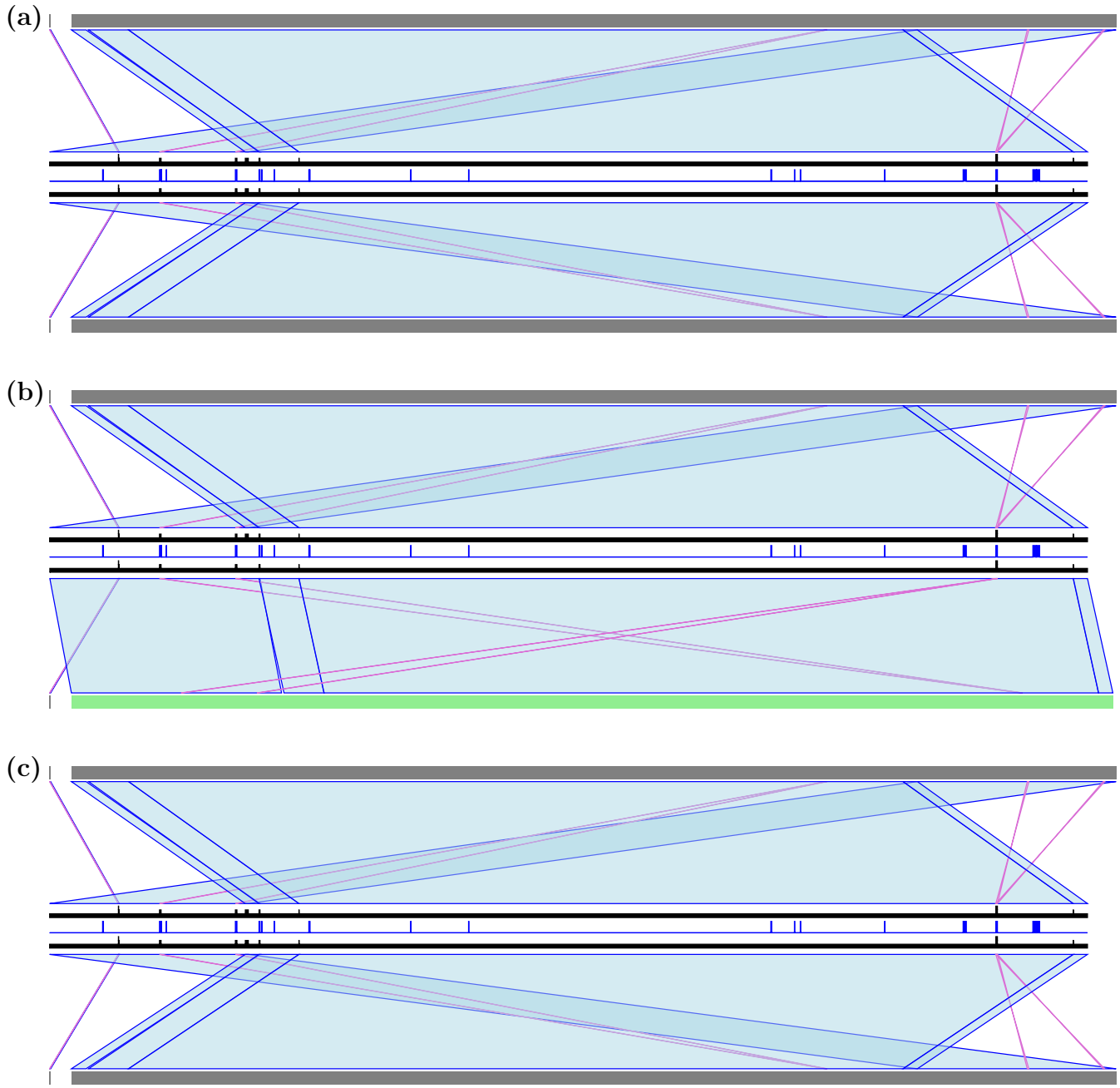

Figure S8: Comparison of circularizing NCTC sample NCTC11192 using (a) BLAST, (b) Circlator and (c) Minimus2.

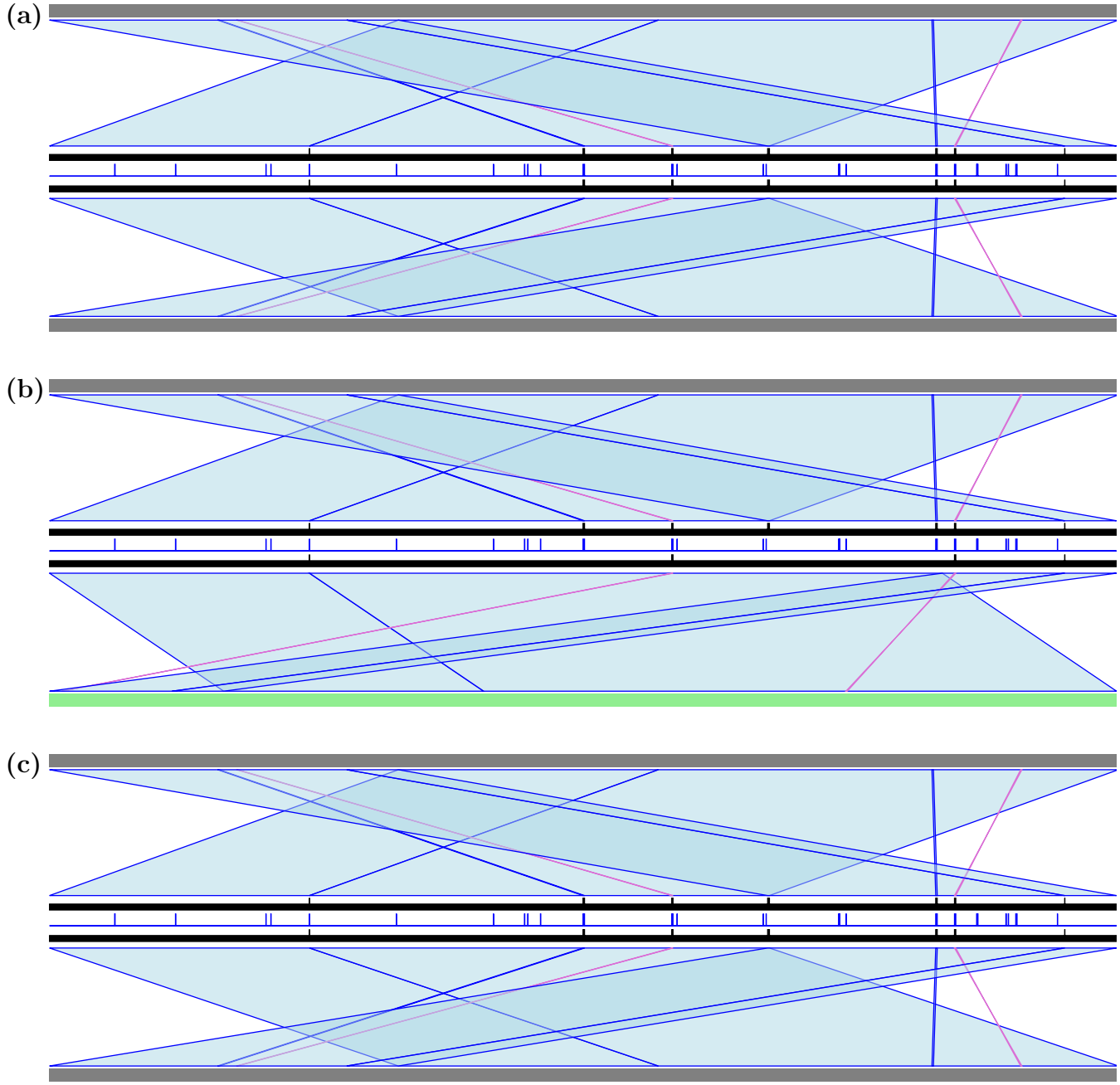

Figure S9: Comparison of circularizing NCTC sample NCTC12419 using (a) BLAST, (b) Circlator and (c) Minimus2.

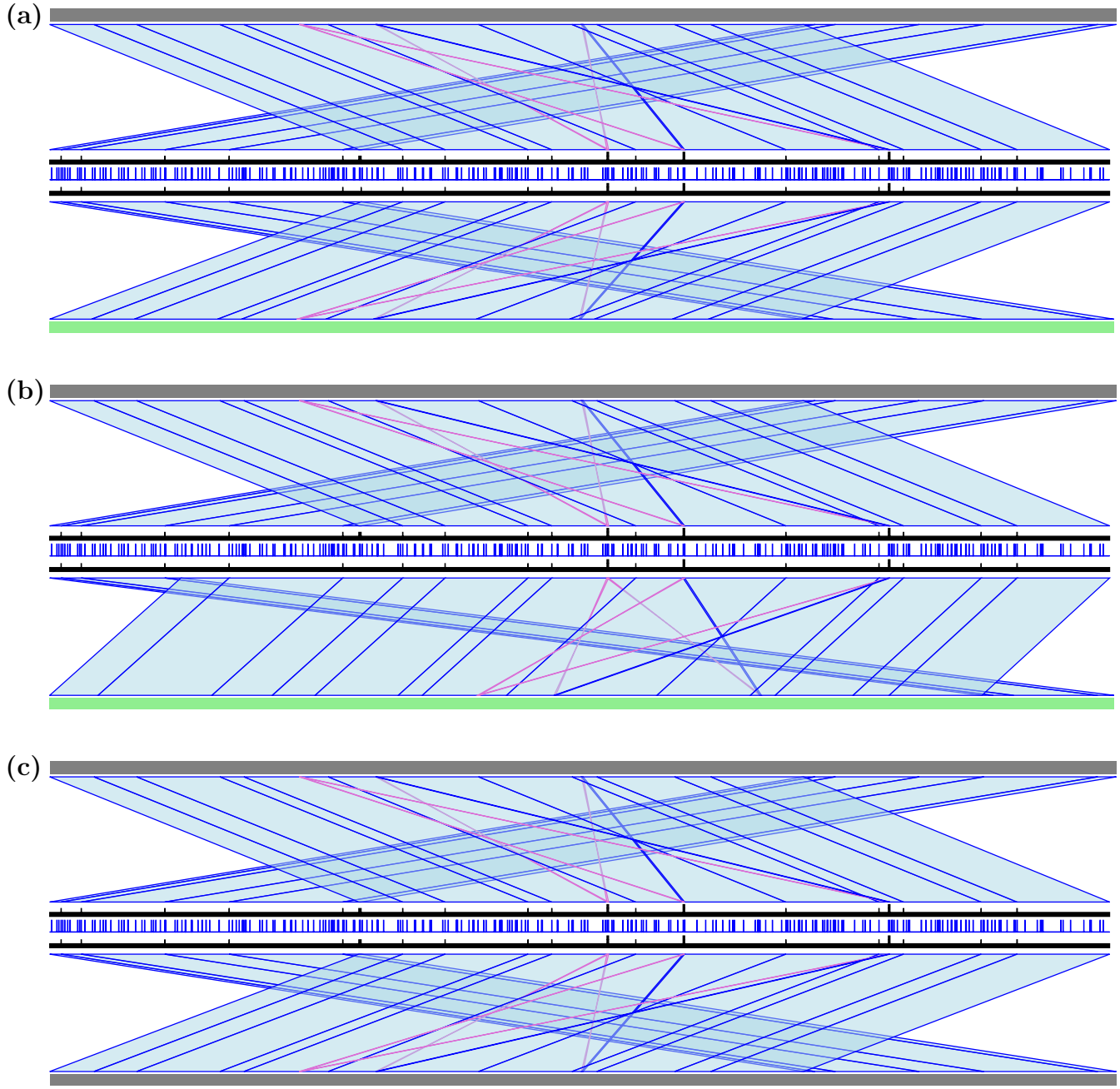

Figure S10: Comparison of circularizing NCTC sample NCTC13251 using (a) BLAST, (b) Circlator and (c) Minimus2.

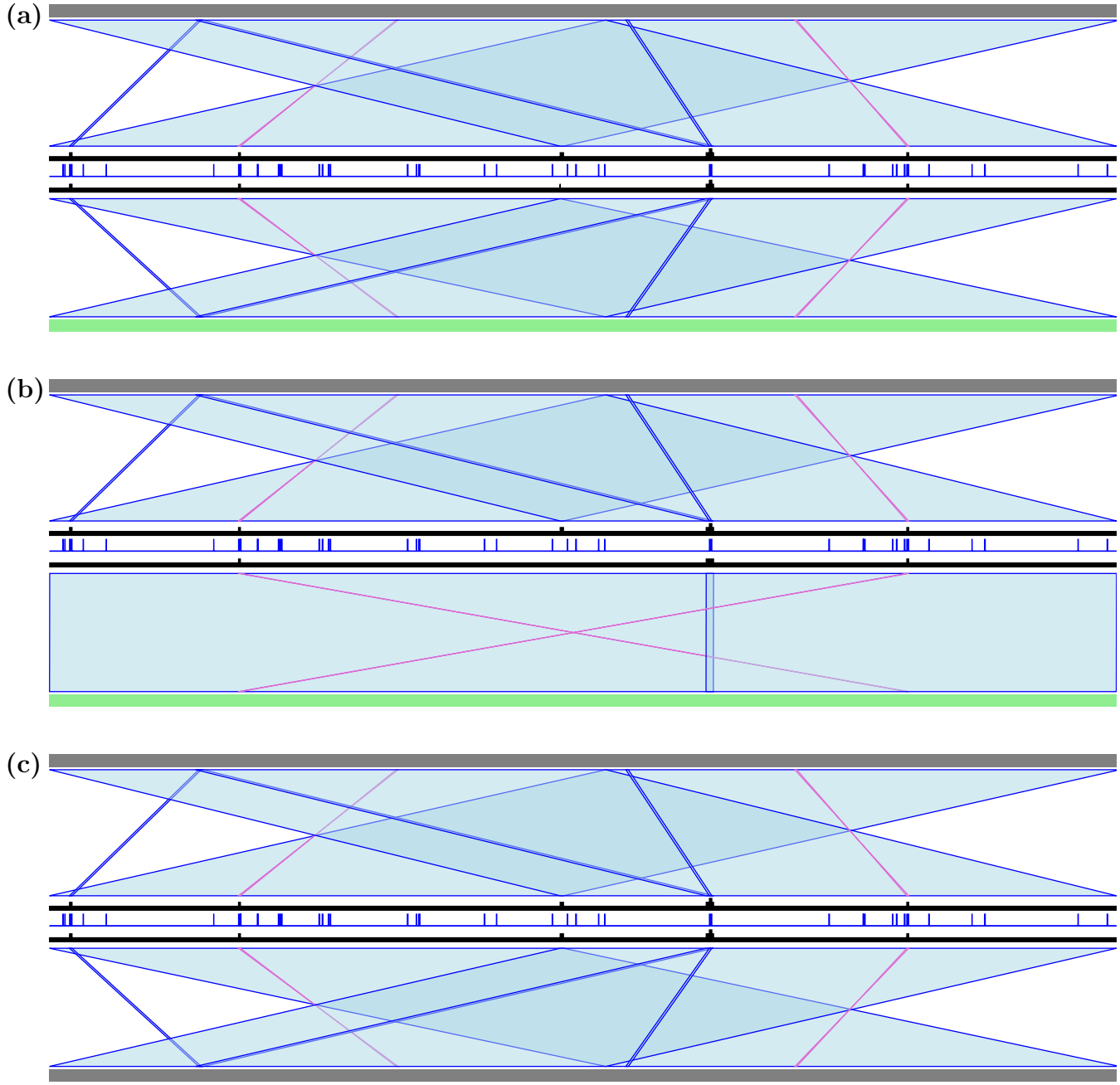

Figure S11: Comparison of circularizing NCTC sample NCTC13277 using (a) BLAST, (b) Circlator and (c) Minimus2.

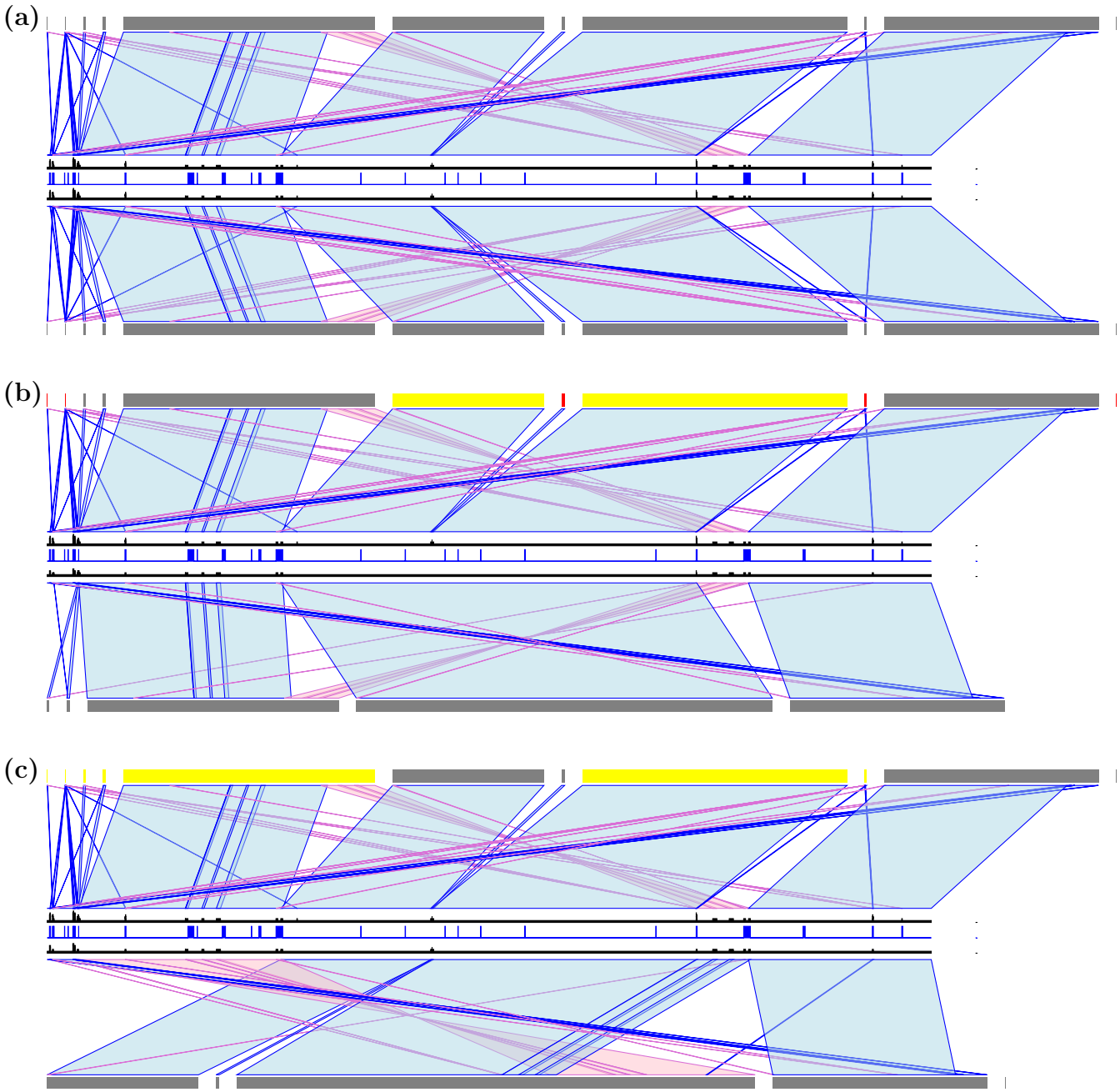

Figure S12: Comparison of circularizing NCTC sample NCTC13307 using (a) BLAST, (b) Circlator and (c) Minimus2.

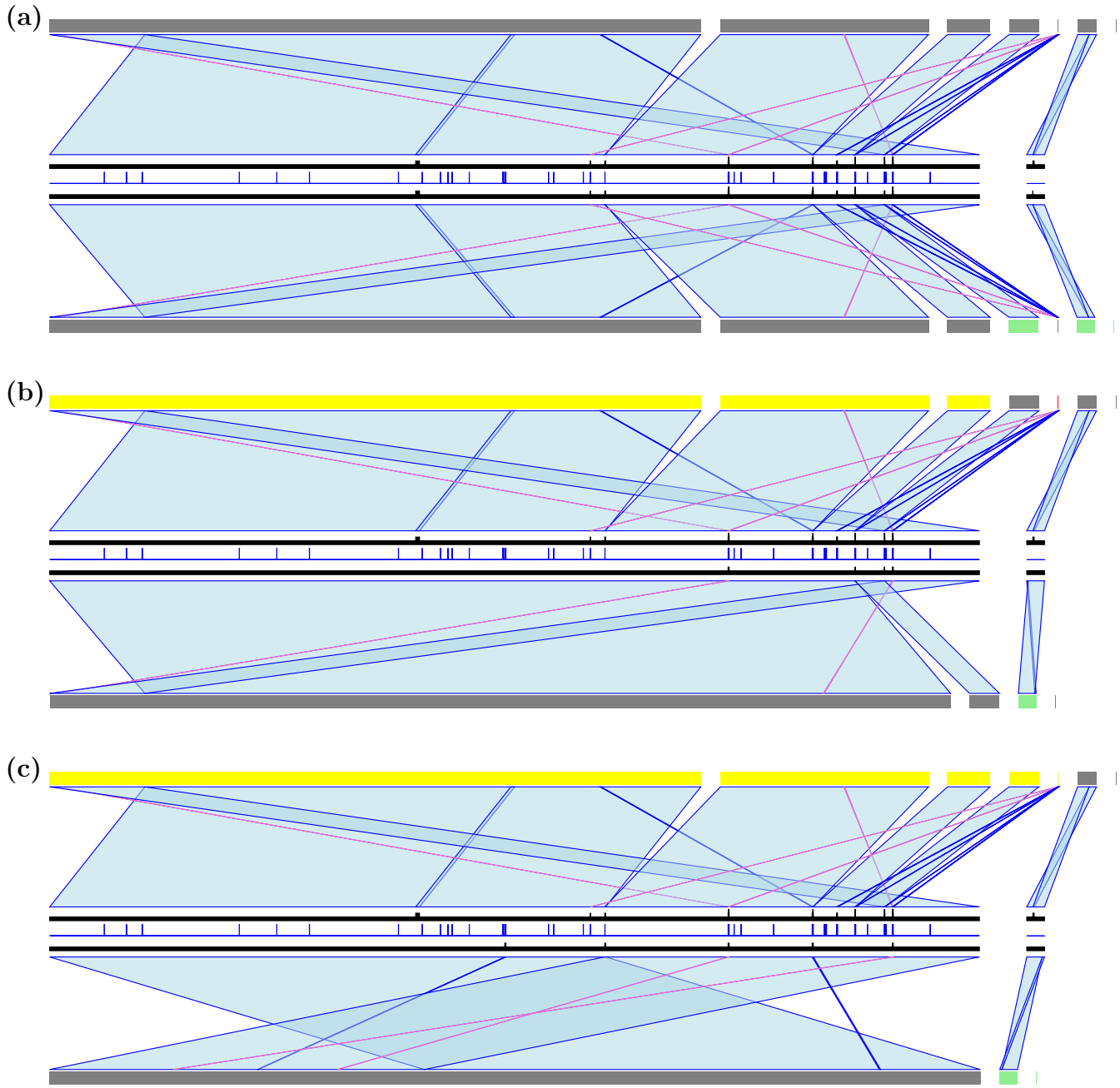

Figure S13: Comparison of circularizing NCTC sample NCTC13348 using (a) BLAST, (b) Circlator and (c) Minimus2.

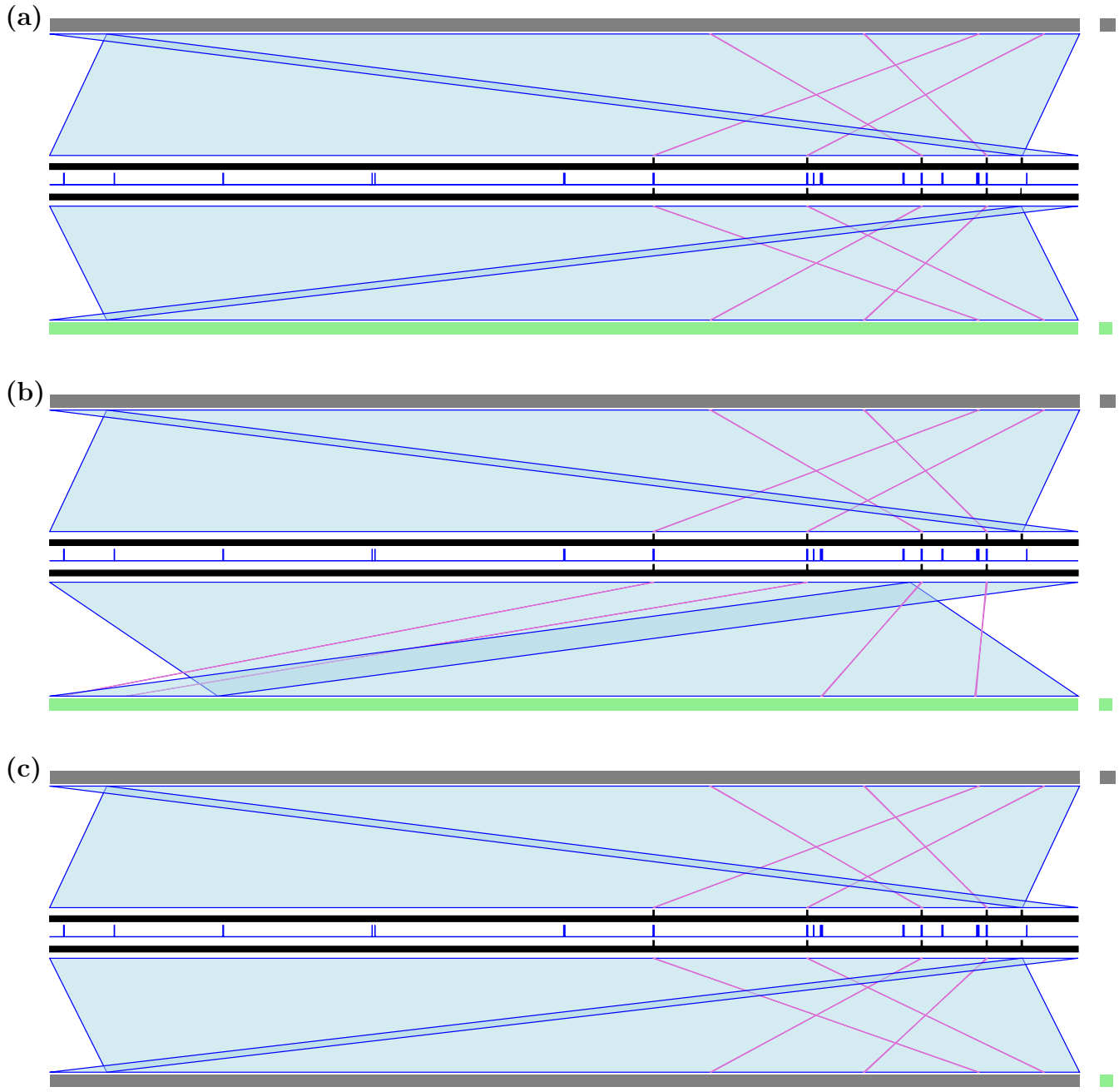

Figure S14: Comparison of circularizing NCTC sample NCTC13349 using (a) BLAST, (b) Circlator and (c) Minimus2.

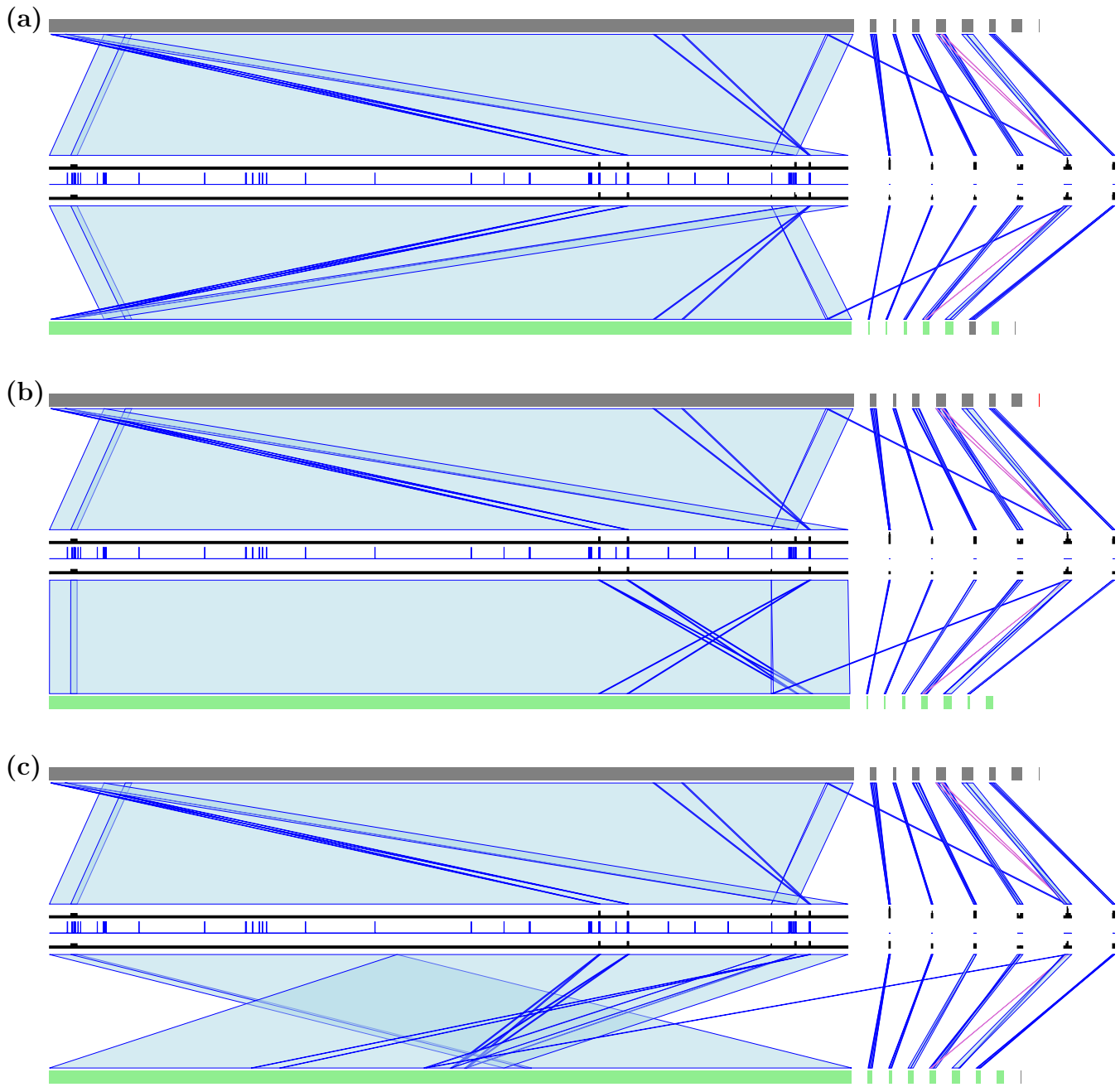

Figure S15: Comparison of circularizing NCTC sample NCTC13360 using (a) BLAST, (b) Circlator and (c) Minimus2.

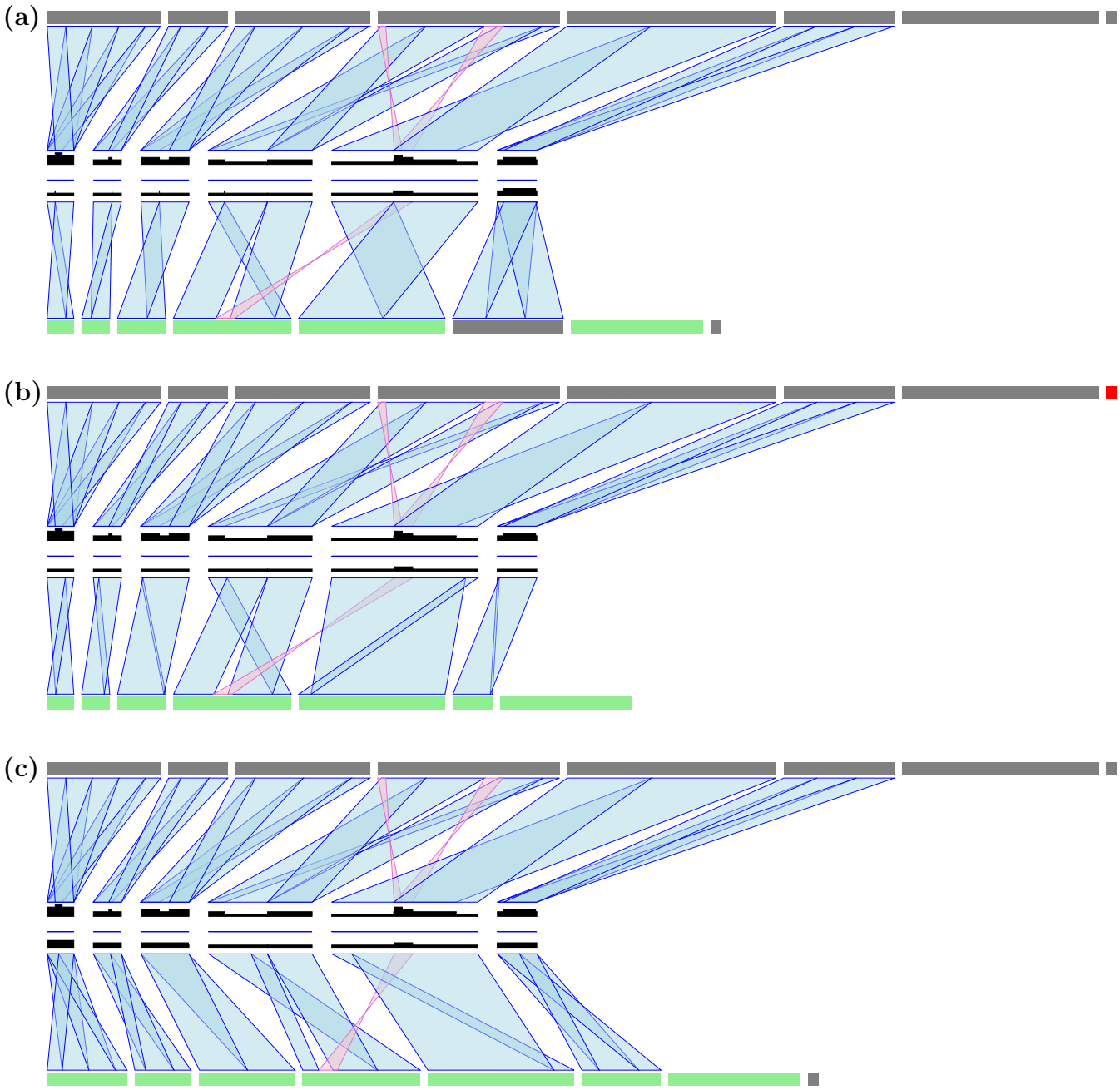

Figure S16: Comparison of circularizing the plasmids of NCTC sample NCTC13360 using (a) BLAST, (b) Circlator and (c) Minimus2.

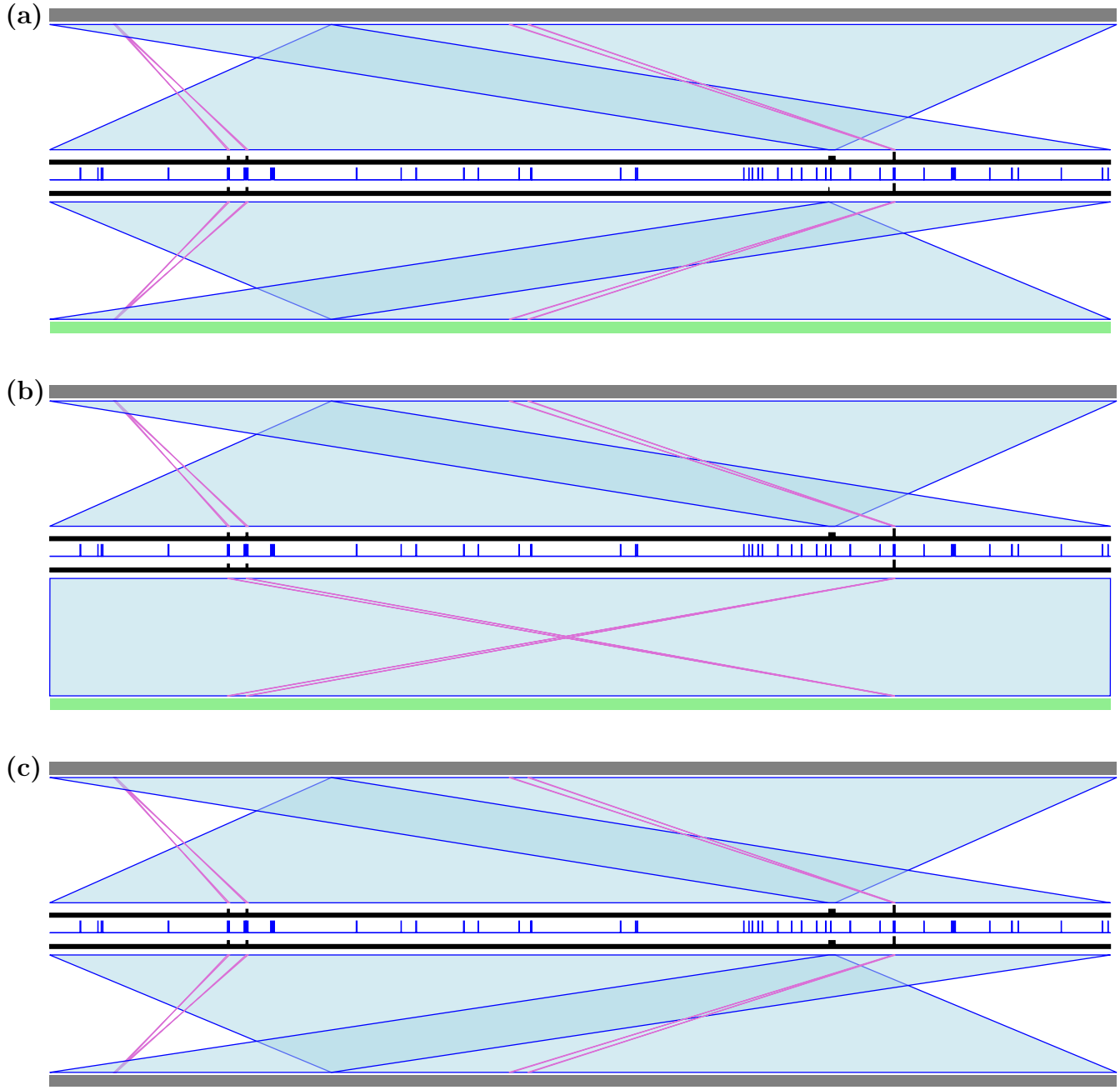

Figure S17: Comparison of circularizing NCTC sample NCTC13616 using (a) BLAST, (b) Circlator and (c) Minimus2.

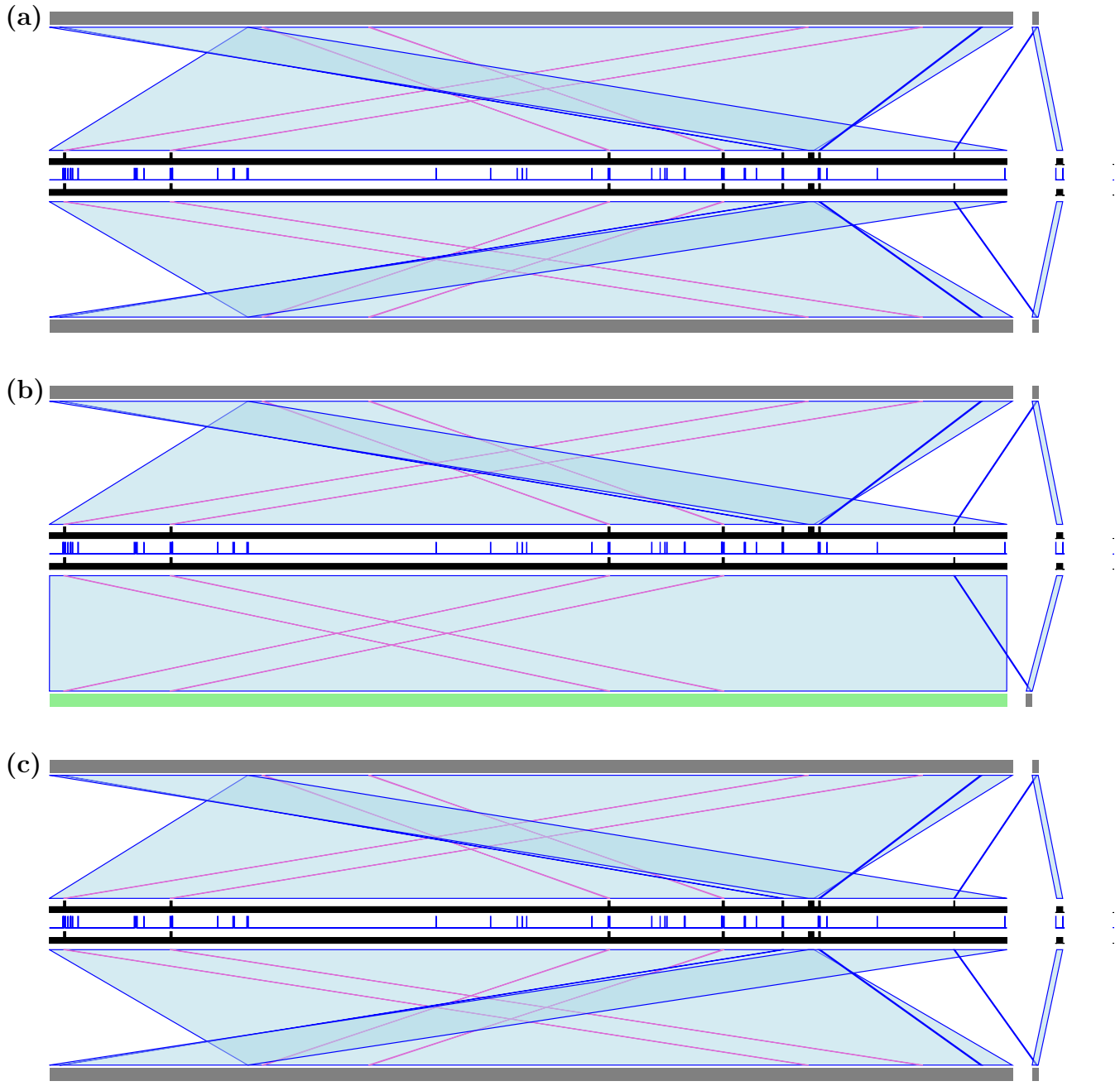

Figure S18: Comparison of circularizing NCTC sample NCTC13626 using (a) BLAST, (b) Circlator and (c) Minimus2.

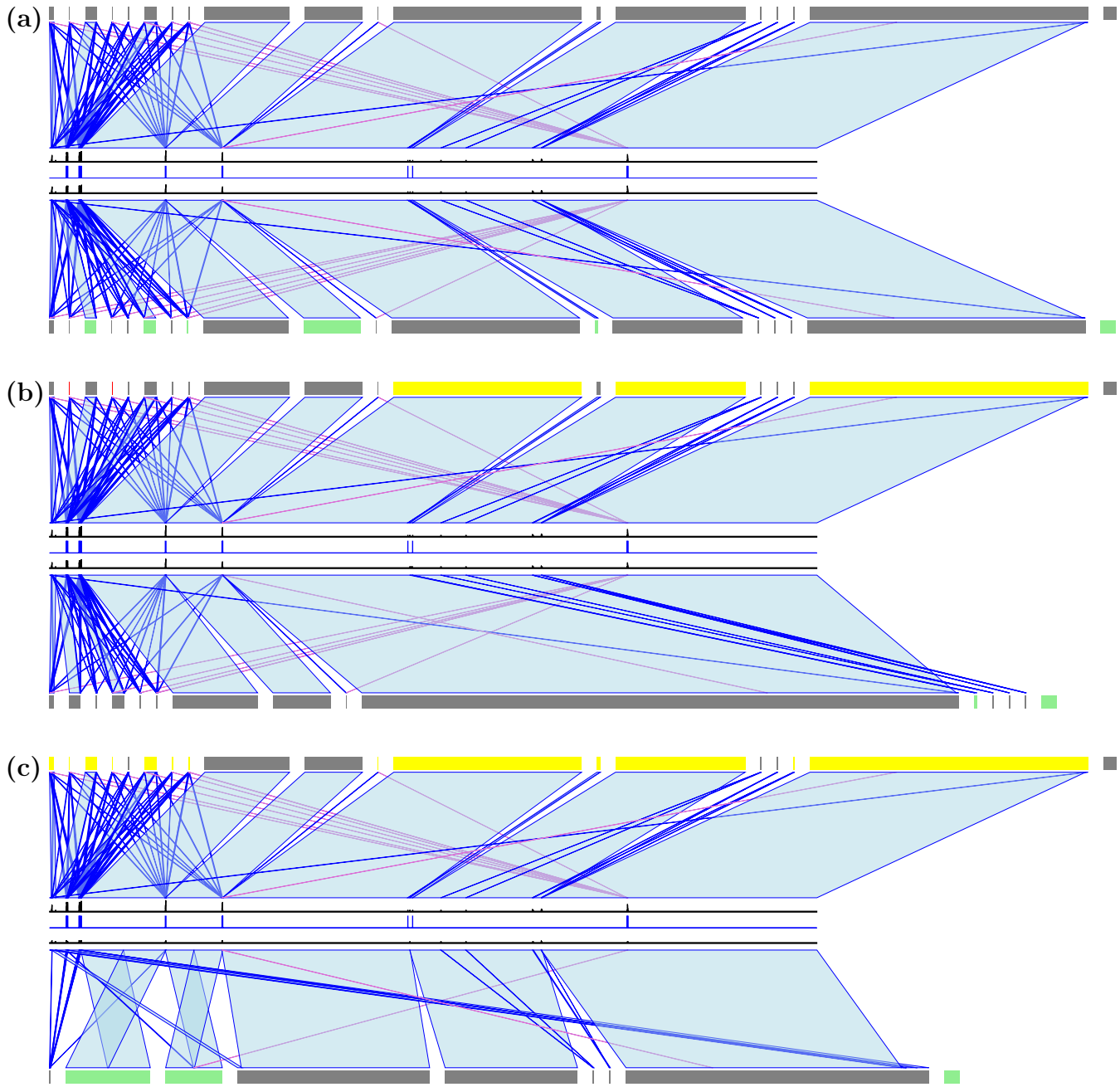

Figure S19: Comparison of circularizing NCTC sample NCTC3610 using (a) BLAST, (b) Circlator and (c) Minimus2.

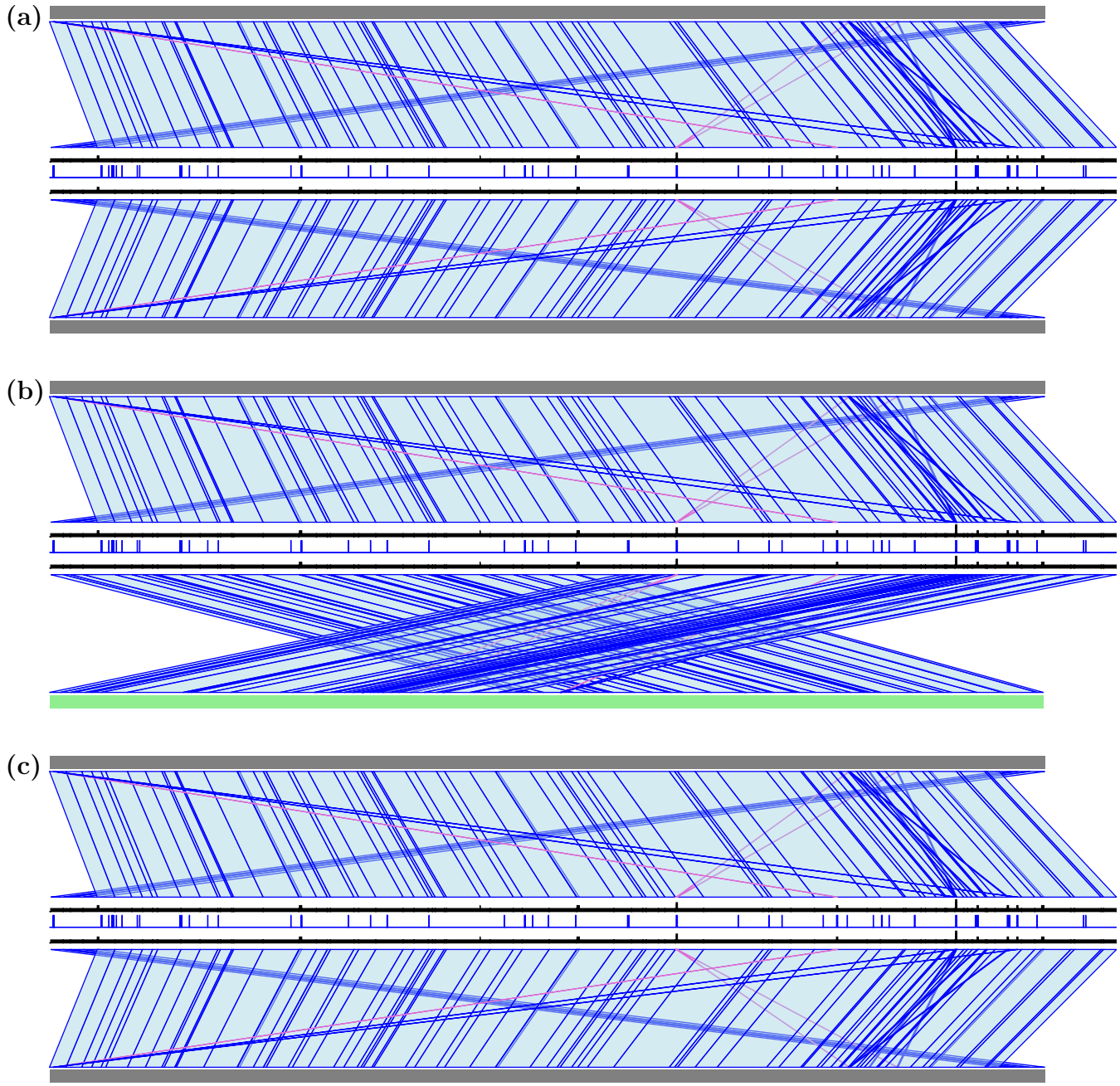

Figure S20: Comparison of circularizing the nanopore assembly using (a) BLAST, (b) Circlator and (c) Minimus2.

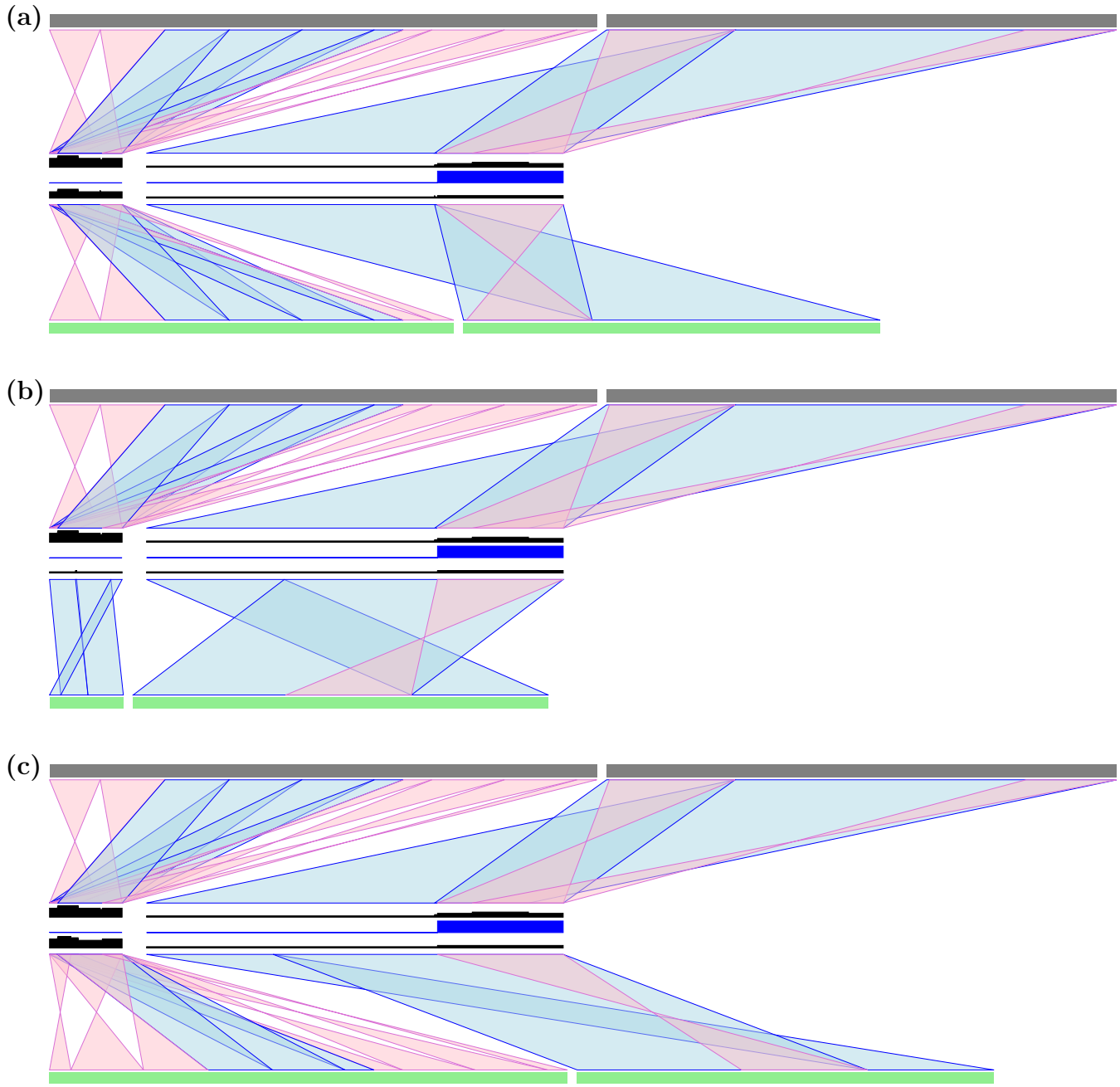

Figure S21: Comparison of circularizing the *P. falciparum* mitochondrion and apicoplast assemblies using (a) BLAST, (b) Circlator and (c) Minimus2.

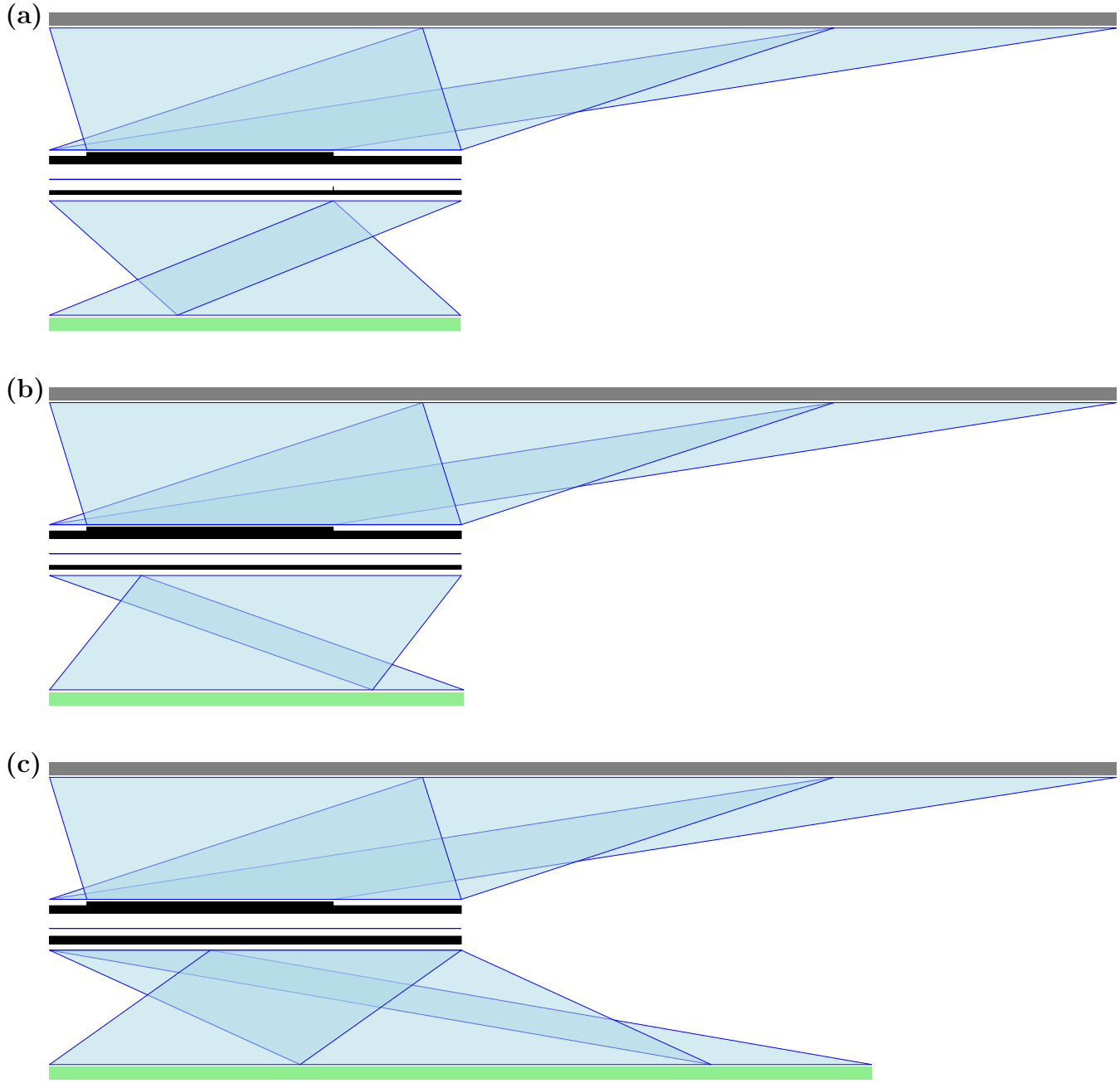

Figure S22: Comparison of circularizing the *H. sapiens* mitochondrion assembly using (a) BLAST, (b) Circlator and (c) Minimus2.

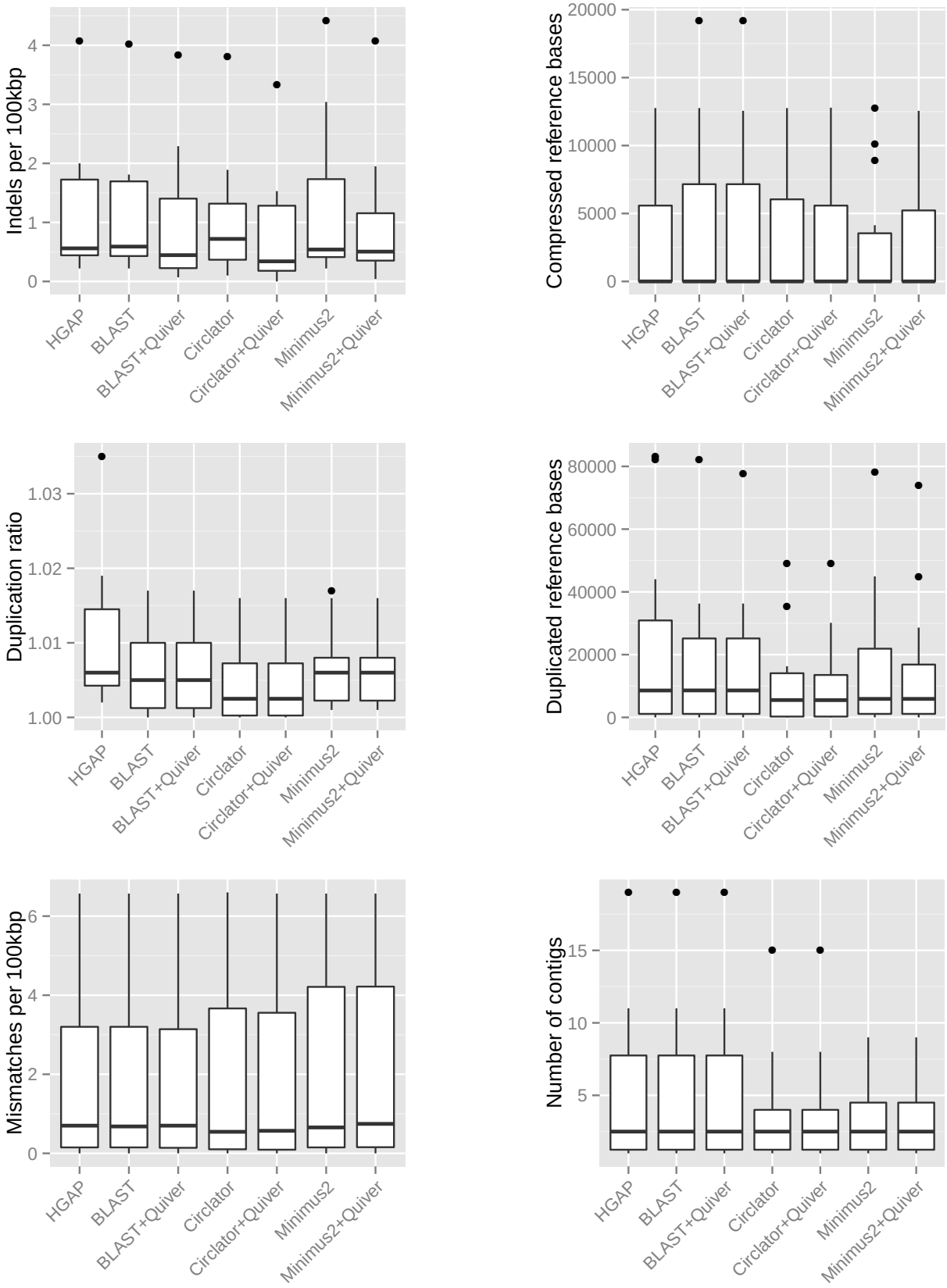

Figure S23: Summary of QUAST statistics on the 14 NCTC samples, comparing the input assemblies, the output of Circlator and the BLAST- and Minmus2-based methods, and the effect of Quiver on the output of those tools.

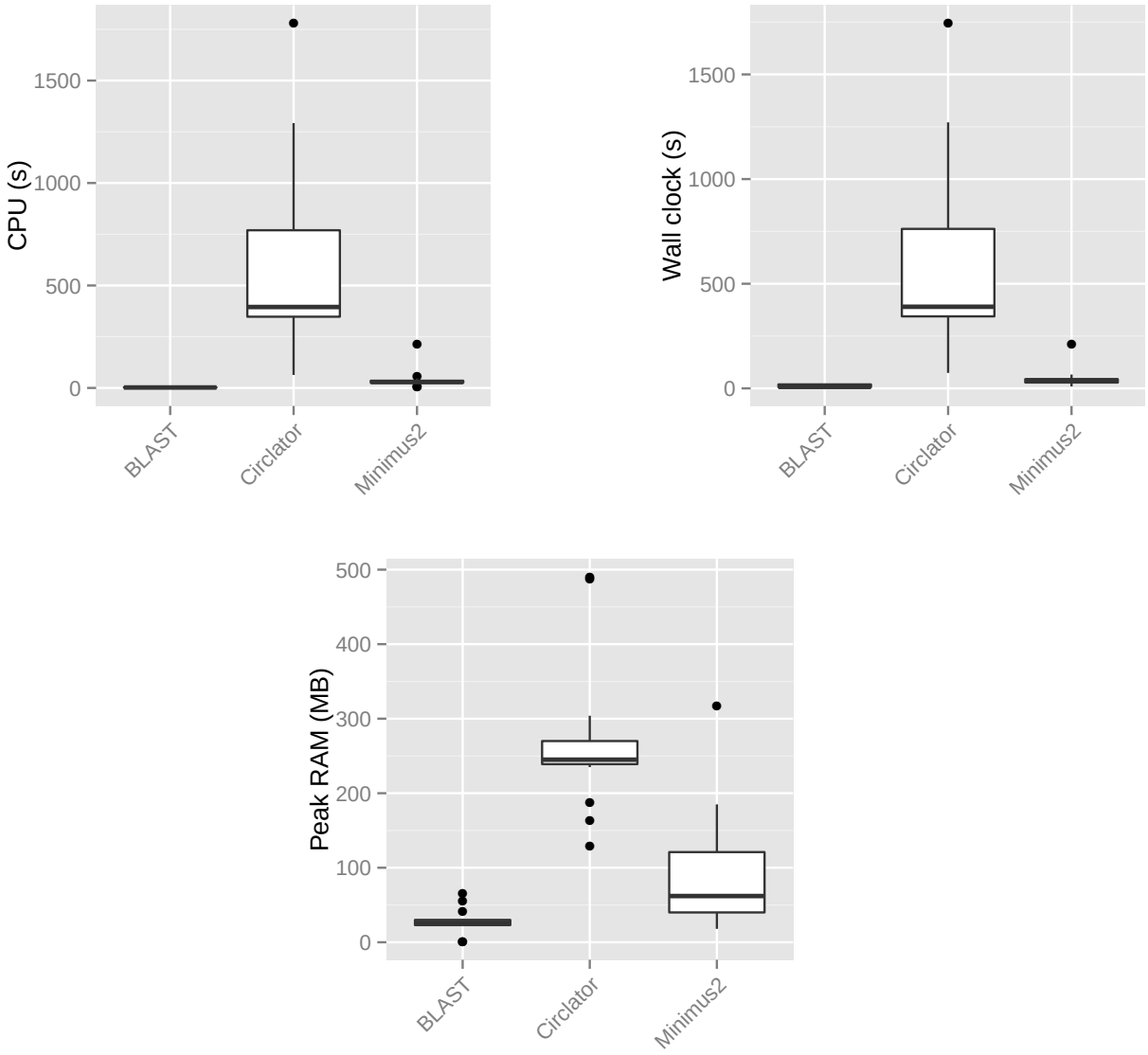

Figure S24: Summary of running time and RAM usage for all datasets

6 References

[1] T. Carver, S. R. Harris, T. D. Otto, M. Berriman, J. Parkhill, and J. A. McQuillan. BamView: visualizing and interpretation of next-generation sequencing read alignments. *Briefings in bioinformatics*, 14(2), Jan. 2012.