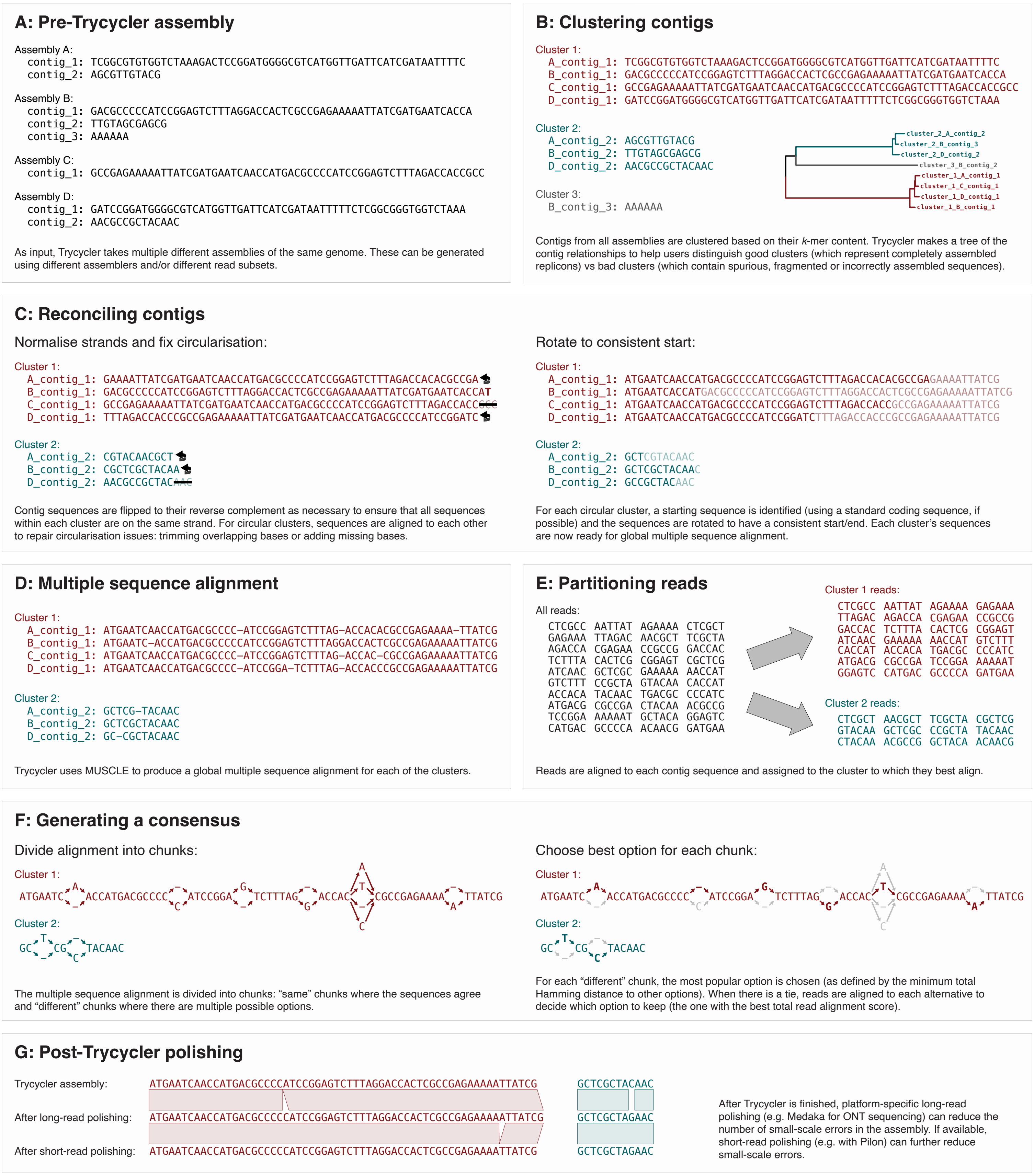


A: Pre-Trycycler assembly

```
Assembly A:
  contig_1: TCGGCGTGTGGTCTAAAGACTCCGGATGGGGCGTCATGGTTGATTCATCGATAATTTTC
  contig_2: AGCGTTGTACG

Assembly B:
  contig_1: GACGCCCCCATCCGGAGTCTTTAGGACCACTCGCCGAGAAAAATTATCGATGAATCACCA
  contig_2: TTGTAGCGAGCG
  contig_3: AAAAAA

Assembly C:
  contig_1: GCCGAGAAAAATTATCGATGAATCAACCATGACGCCCCATCCGGAGTCTTTAGACCACCGCC

Assembly D:
  contig_1: GATCCGGATGGGGCGTCATGGTTGATTCATCGATAATTTTTCTCGGCGGGTGGTCTAAA
  contig_2: AACGCCGCTACAAC
```

As input, Trycycler takes multiple different assemblies of the same genome. These can be generated using different assemblers and/or different read subsets.

B: Clustering contigs

```
Cluster 1:
  A_contig_1: TCGGCGTGTGGTCTAAAGACTCCGGATGGGGCGTCATGGTTGATTCATCGATAATTTTC
  B_contig_1: GACGCCCCCATCCGGAGTCTTTAGGACCACTCGCCGAGAAAAATTATCGATGAATCACCA
  C_contig_1: GCCGAGAAAAATTATCGATGAATCAACCATGACGCCCCATCCGGAGTCTTTAGACCACCGCC
  D_contig_1: GATCCGGATGGGGCGTCATGGTTGATTCATCGATAATTTTTCTCGGCGGGTGGTCTAAA

Cluster 2:
  A_contig_2: AGCGTTGTACG
  B_contig_2: TTGTAGCGAGCG
  D_contig_2: AACGCCGCTACAAC

Cluster 3:
  B_contig_3: AAAAAA
```

Contigs from all assemblies are clustered based on their *k*-mer content. Trycycler makes a tree of the contig relationships to help users distinguish good clusters (which represent completely assembled replicons) vs bad clusters (which contain spurious, fragmented or incorrectly assembled sequences).

C: Reconciling contigs

Normalise strands and fix circularisation:

```
Cluster 1:
  A_contig_1: GAAAATTATCGATGAATCAACCATGACGCCCCATCCGGAGTCTTTAGACCACACGCCGA
  B_contig_1: GACGCCCCCATCCGGAGTCTTTAGGACCACTCGCCGAGAAAAATTATCGATGAATCACCAT
  C_contig_1: GCCGAGAAAAATTATCGATGAATCAACCATGACGCCCCATCCGGAGTCTTTAGACCACC~~GCC~~
  D_contig_1: TTTAGACCACCCGCCGAGAAAAATTATCGATGAATCAACCATGACGCCCCATCCGGATC

Cluster 2:
  A_contig_2: CGTACAACGCT
  B_contig_2: CGCTCGCTACAA
  D_contig_2: AACGCCGCTAC~~AAC~~
```

Contig sequences are flipped to their reverse complement as necessary to ensure that all sequences within each cluster are on the same strand. For circular clusters, sequences are aligned to each other to repair circularisation issues: trimming overlapping bases or adding missing bases.

Rotate to consistent start:

```
Cluster 1:
  A_contig_1: ATGAATCAACCATGACGCCCCATCCGGAGTCTTTAGACCACACGCCGAGAAAATTATCG
  B_contig_1: ATGAATCACCATGACGCCCCCATCCGGAGTCTTTAGGACCACTCGCCGAGAAAAATTATCG
  C_contig_1: ATGAATCAACCATGACGCCCCATCCGGAGTCTTTAGACCACCGCCGAGAAAAATTATCG
  D_contig_1: ATGAATCAACCATGACGCCCCATCCGGATCTTTAGACCACCCGCCGAGAAAAATTATCG

Cluster 2:
  A_contig_2: GCTCGTACAAC
  B_contig_2: GCTCGCTACAAC
  D_contig_2: GCCGCTACAAC
```

For each circular cluster, a starting sequence is identified (using a standard coding sequence, if possible) and the sequences are rotated to have a consistent start/end. Each cluster's sequences are now ready for global multiple sequence alignment.

D: Multiple sequence alignment

```
Cluster 1:
  A_contig_1: ATGAATCAACCATGACGCCCC-ATCCGGAGTCTTTAG-ACCACACGCCGAGAAAA-TTATCG
  B_contig_1: ATGAATC-ACCATGACGCCCCCATCCGGAGTCTTTAGGACCACTCGCCGAGAAAAATTATCG
  C_contig_1: ATGAATCAACCATGACGCCCC-ATCCGGAGTCTTTAG-ACCAC-CGCCGAGAAAAATTATCG
  D_contig_1: ATGAATCAACCATGACGCCCC-ATCCGGA-TCTTTAG-ACCACCCGCCGAGAAAAATTATCG

Cluster 2:
  A_contig_2: GCTCG-TACAAC
  B_contig_2: GCTCGCTACAAC
  D_contig_2: GC-CGCTACAAC
```

Trycycler uses MUSCLE to produce a global multiple sequence alignment for each of the clusters.

E: Partitioning reads

```
All reads:
  CTCGCC  AATTAT  AGAAAA  CTCGCT
  GAGAAA  TTAGAC  AACGCT  TCGCTA
  AGACCA  CGAGAA  CCGCCG  GACCAC
  TCTTTA  CACTCG  CGGAGT  CGCTCG
  ATCAAC  GCTCGC  GAAAAA  AACCAT
  GTCTTT  CCGCTA  GTACAA  CACCAT
  ACCACA  TACAAC  TGACGC  CCCATC
  ATGACG  CGCCGA  CTACAA  ACGCCG
  TCCGGA  AAAAAT  GCTACA  GGAGTC
  CATGAC  GCCCCA  ACAACG  GATGAA

Cluster 1 reads:
  CTCGCC  AATTAT  AGAAAA  GAGAAA
  TTAGAC  AGACCA  CGAGAA  CCGCCG
  GACCAC  TCTTTA  CACTCG  CGGAGT
  ATCAAC  GAAAAA  AACCAT  GTCTTT
  CACCAT  ACCACA  TGACGC  CCCATC
  ATGACG  CGCCGA  TCCGGA  AAAAAT
  GGAGTC  CATGAC  GCCCCA  GATGAA

Cluster 2 reads:
  CTCGCT  AACGCT  TCGCTA  CGCTCG
  GTACAA  GCTCGC  CCGCTA  TACAAC
  CTACAA  ACGCCG  GCTACA  ACAACG
```

Reads are aligned to each contig sequence and assigned to the cluster to which they best align.

F: Generating a consensus

Divide alignment into chunks:

Cluster 1:

Cluster 2:

The multiple sequence alignment is divided into chunks: "same" chunks where the sequences agree and "different" chunks where there are multiple possible options.

Choose best option for each chunk:

Cluster 1:

Cluster 2:

For each "different" chunk, the most popular option is chosen (as defined by the minimum total Hamming distance to other options). When there is a tie, reads are aligned to each alternative to decide which option to keep (the one with the best total read alignment score).

G: Post-Trycycler polishing

```
Trycycler assembly:         ATGAATCAACCATGACGCCCCATCCGGAGTCTTTAGGACCACTCGCCGAGAAAAATTATCG     GCTCGCTACAAC

After long-read polishing:  ATGAATCAACCATGACGCCCCCATCCGGAGTCTTTAGGACCACTCGCCGAGAAAAATTATCG    GCTCGCTAGAAC

After short-read polishing: ATGAATCAACCATGACGCCCCCATCCGGAGTCTTTAGGACCACTCGCCGAGAAAATTATCG     GCTCGCTAGAAC
```

After Trycycler is finished, platform-specific long-read polishing (e.g. Medaka for ONT sequencing) can reduce the number of small-scale errors in the assembly. If available, short-read polishing (e.g. with Pilon) can further reduce small-scale errors.

Fig. S1: steps in the Trycycler assembly pipeline.

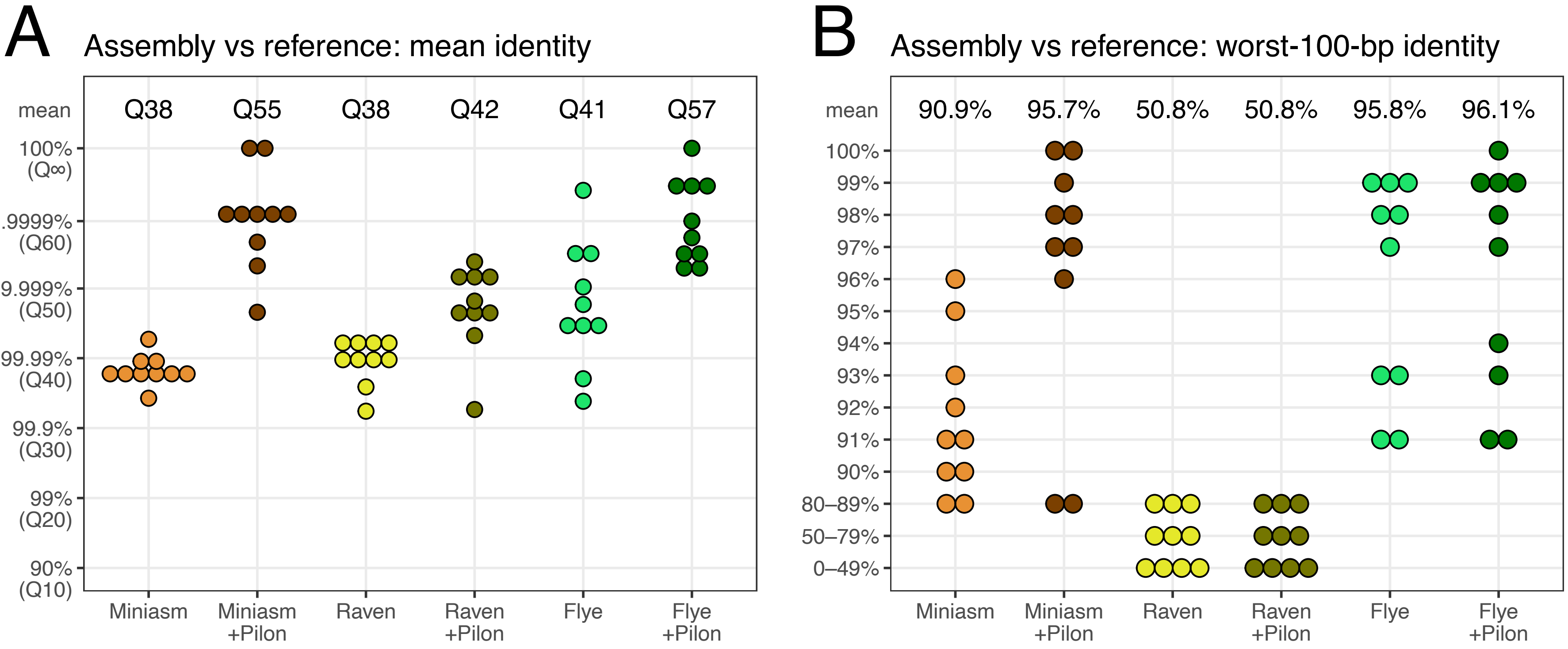


Fig. S2: results for the simulated read tests for long-read-only assemblers. This figure contains the same analyses as are shown in Figure 2, but it includes assemblies from all long-read-only assemblers (both before and after Pilon polishing).

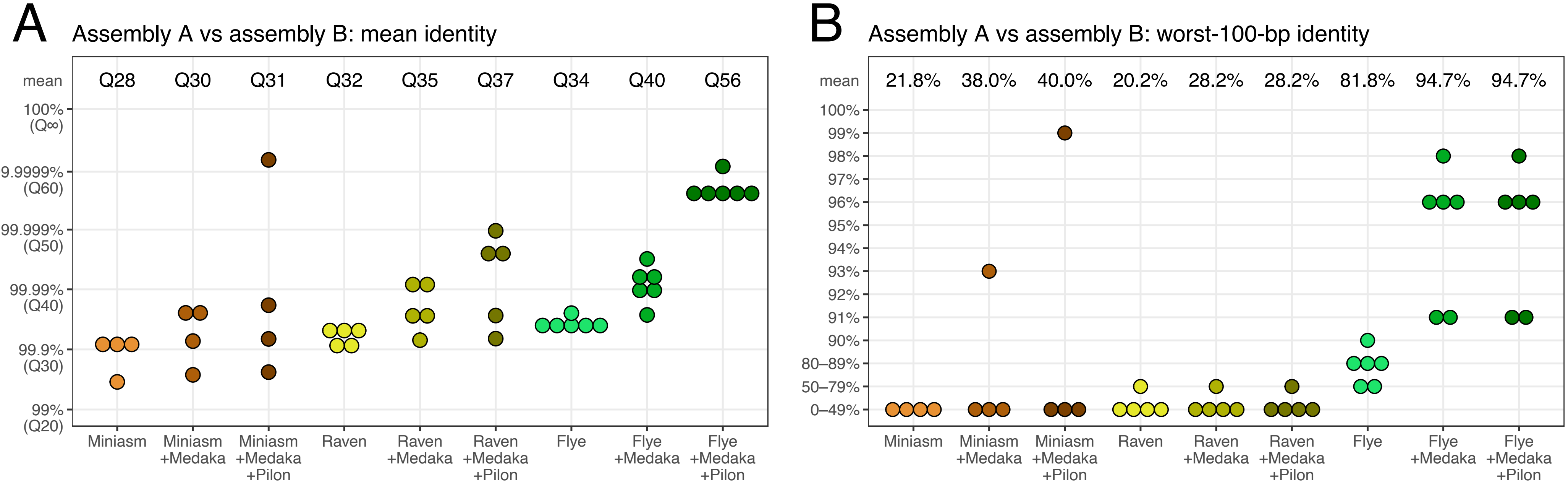


Fig. S3: results for the real read tests for long-read-only assemblers. This figure contains the same analyses as are shown in Figure 3, but it includes assemblies from all long-read-only assemblers (unpolished, Medaka-polished and Medaka+Pilon-polished for each). In two of the six genomes, Miniasm failed to produce a completed chromosome for both read sets, resulting in only four data points. In one of the six genomes, Raven failed to produce a completed chromosome for both read sets, resulting in only five data points.

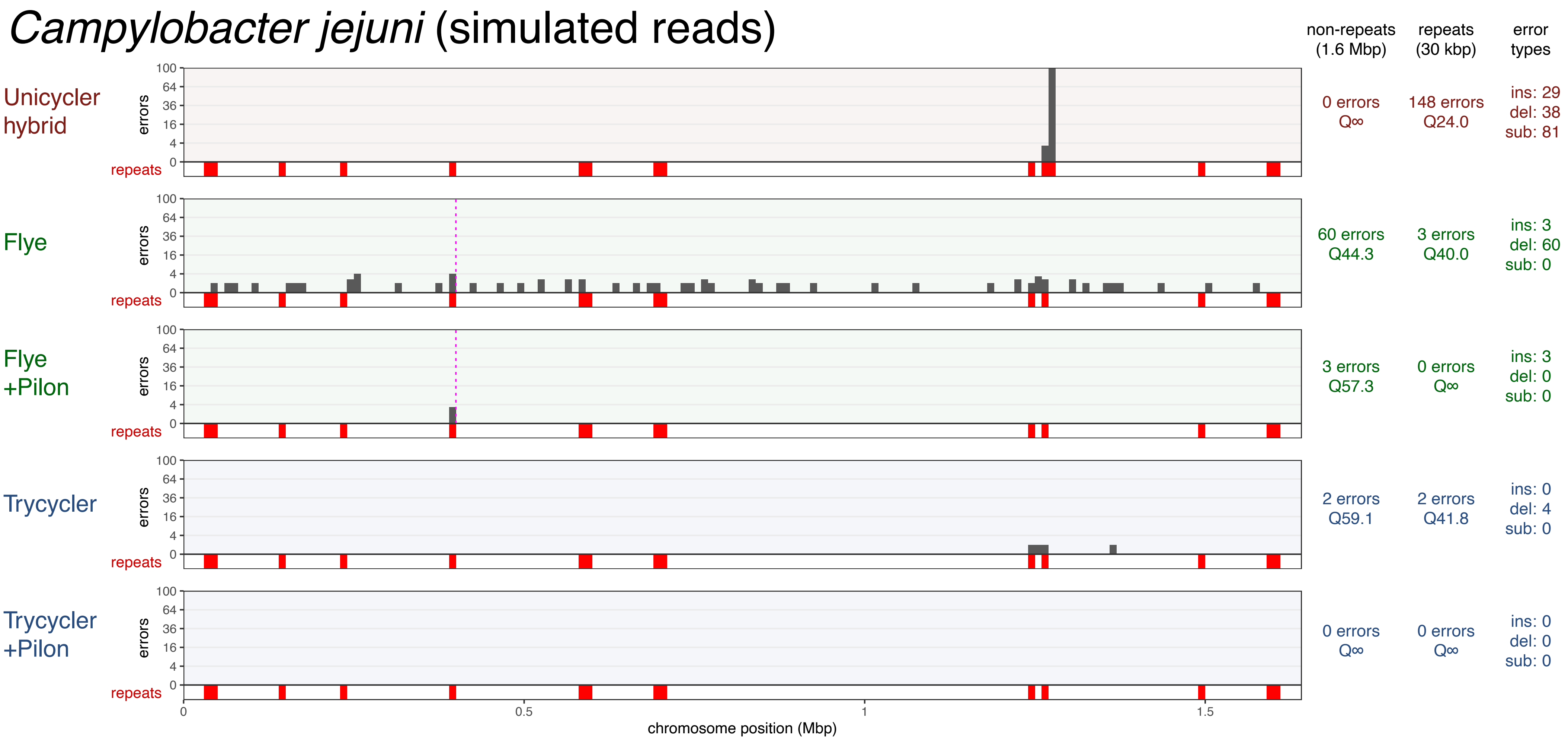
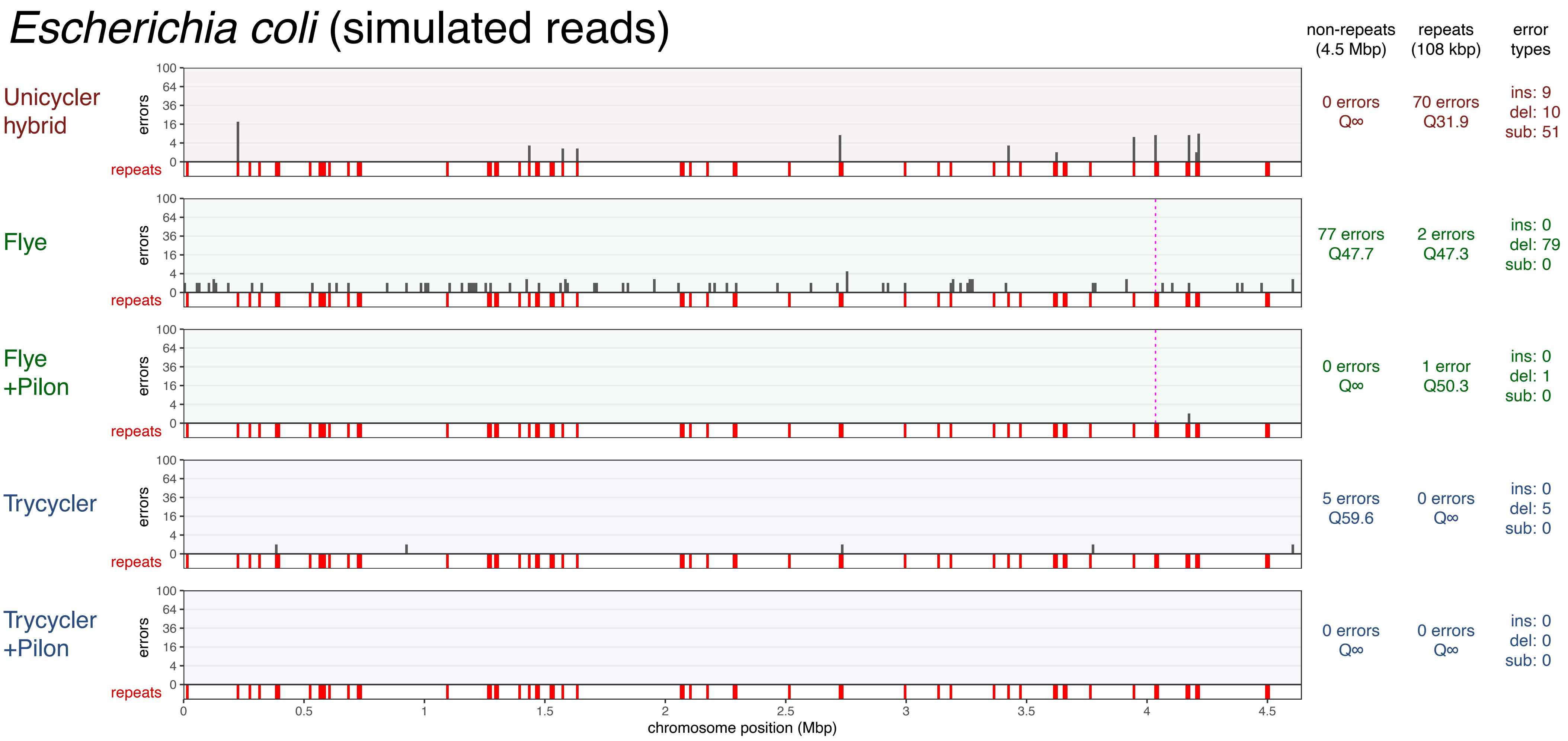
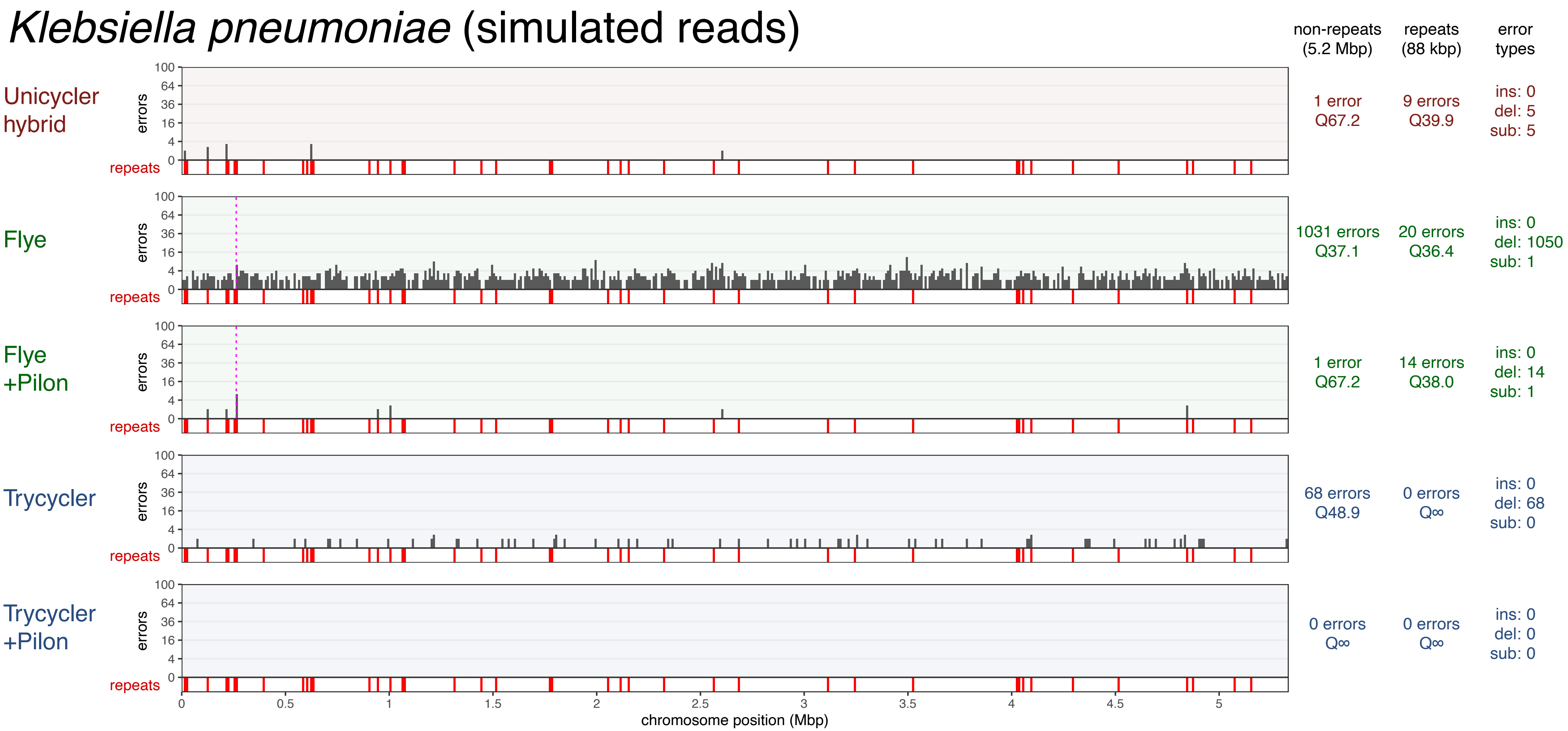


Fig. S4-a: error distributions (as defined by alignment to the reference sequence) for the simulated-read assemblies of the *Campylobacter jejuni* chromosome. Repetitive regions of the genome are shown as red blocks at the bottom of each plot. For Flye assemblies, the contig start/end position (where the sequence began before it was rotated to be consistent with the reference) is shown by the dashed line.



	non-repeats (4.5 Mbp)	repeats (108 kbp)	error types
Unicycler hybrid	0 errors Q∞	70 errors Q31.9	ins: 9 del: 10 sub: 51
Flye	77 errors Q47.7	2 errors Q47.3	ins: 0 del: 79 sub: 0
Flye +Pilon	0 errors Q∞	1 error Q50.3	ins: 0 del: 1 sub: 0
Trycycler	5 errors Q59.6	0 errors Q∞	ins: 0 del: 5 sub: 0
Trycycler +Pilon	0 errors Q∞	0 errors Q∞	ins: 0 del: 0 sub: 0

Fig. S4-b: error distributions (as defined by alignment to the reference sequence) for the simulated-read assemblies of the *Escherichia coli* chromosome. Repetitive regions of the genome are shown as red blocks at the bottom of each plot. For Flye assemblies, the contig start/end position (where the sequence began before it was rotated to be consistent with the reference) is shown by the dashed line.



	non-repeats (5.2 Mbp)	repeats (88 kbp)	error types
Unicycler hybrid	1 error Q67.2	9 errors Q39.9	ins: 0 del: 5 sub: 5
Flye	1031 errors Q37.1	20 errors Q36.4	ins: 0 del: 1050 sub: 1
Flye +Pilon	1 error Q67.2	14 errors Q38.0	ins: 0 del: 14 sub: 1
Trycycler	68 errors Q48.9	0 errors Q∞	ins: 0 del: 68 sub: 0
Trycycler +Pilon	0 errors Q∞	0 errors Q∞	ins: 0 del: 0 sub: 0

Fig. S4-c: error distributions (as defined by alignment to the reference sequence) for the simulated-read assemblies of the *Klebsiella pneumoniae* chromosome. Repetitive regions of the genome are shown as red blocks at the bottom of each plot. For Flye assemblies, the contig start/end position (where the sequence began before it was rotated to be consistent with the reference) is shown by the dashed line.

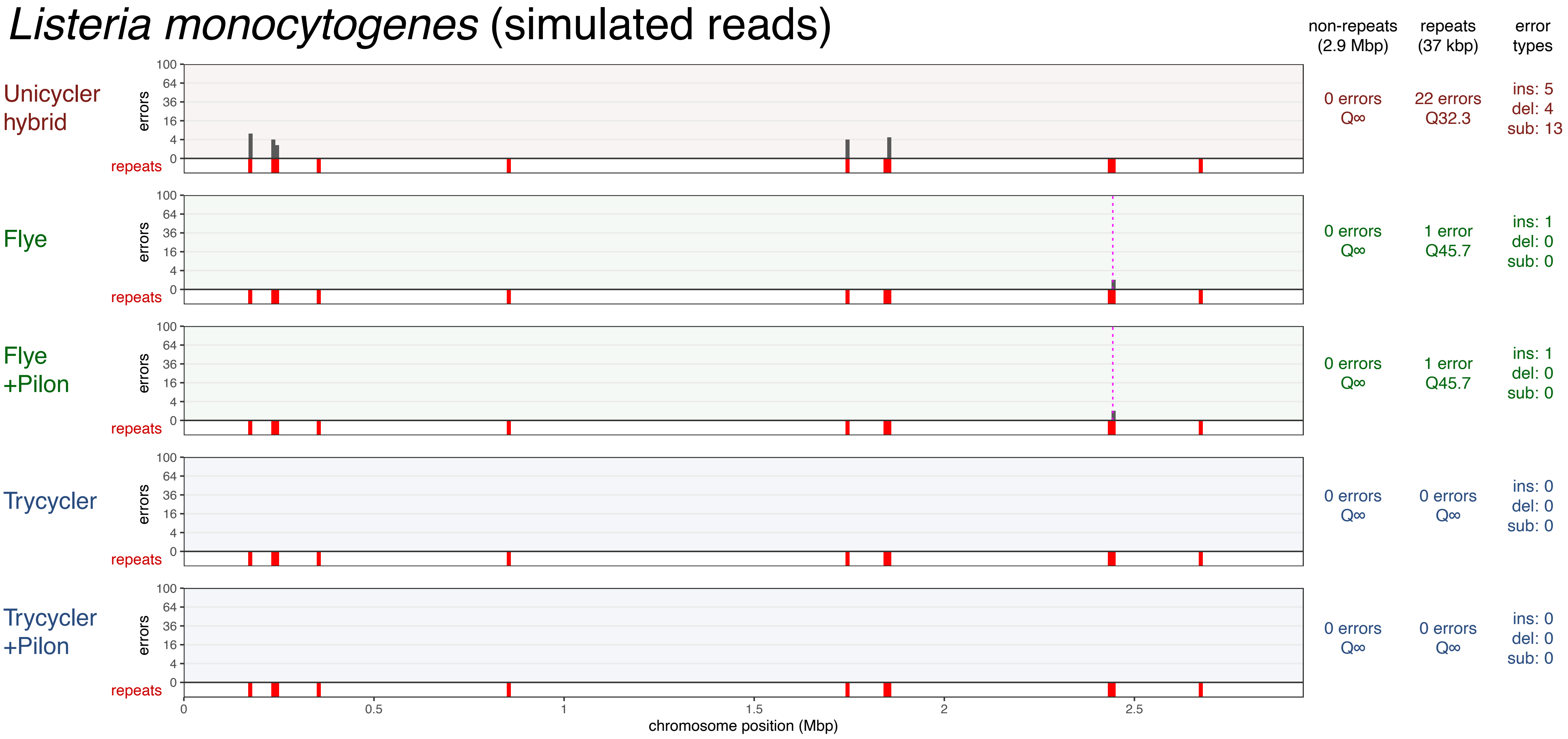


Fig. S4-d: error distributions (as defined by alignment to the reference sequence) for the simulated-read assemblies of the *Listeria monocytogenes* chromosome. Repetitive regions of the genome are shown as red blocks at the bottom of each plot. For Flye assemblies, the contig start/end position (where the sequence began before it was rotated to be consistent with the reference) is shown by the dashed line.

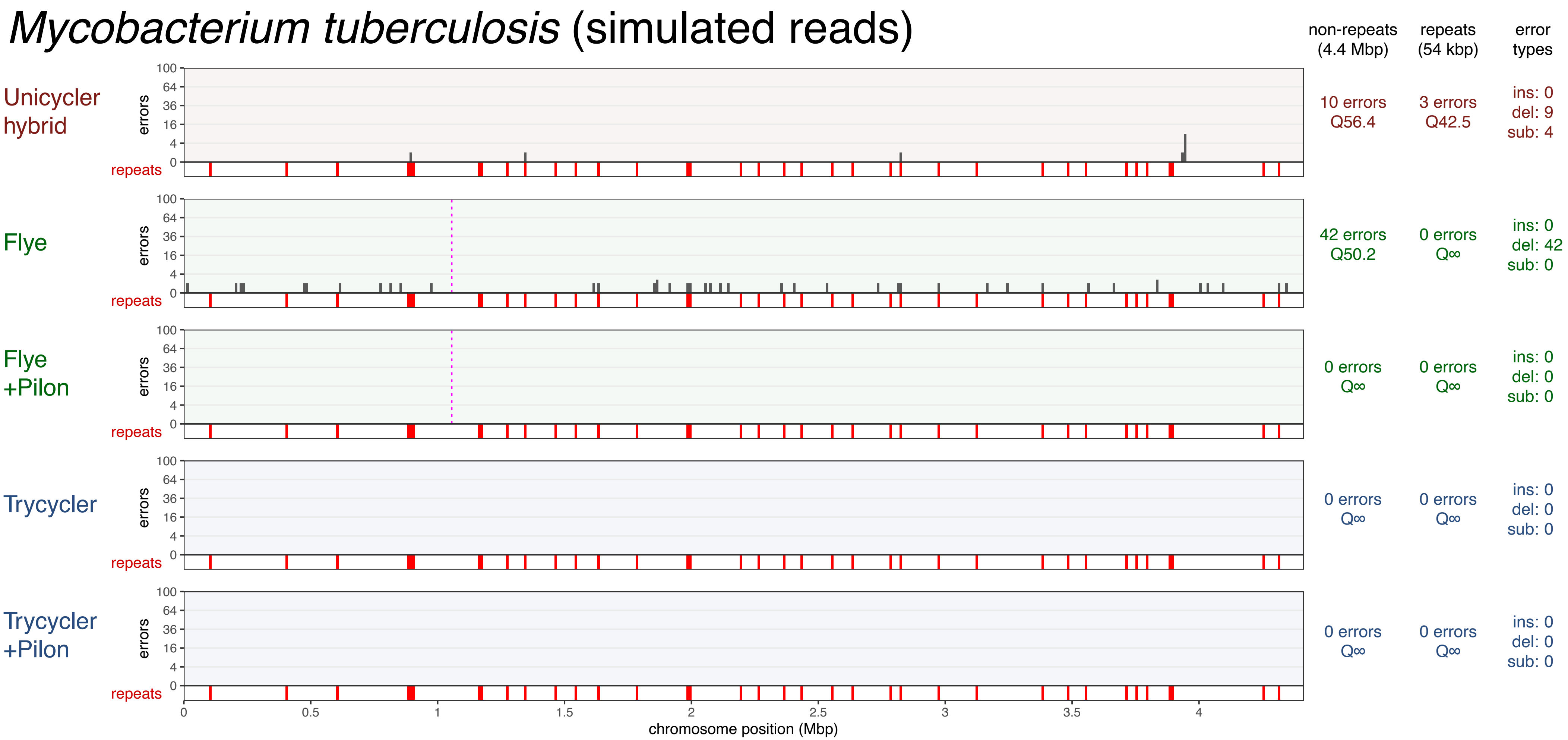


Fig. S4-e: error distributions (as defined by alignment to the reference sequence) for the simulated-read assemblies of the *Mycobacterium tuberculosis* chromosome. Repetitive regions of the genome are shown as red blocks at the bottom of each plot. For Flye assemblies, the contig start/end position (where the sequence began before it was rotated to be consistent with the reference) is shown by the dashed line.

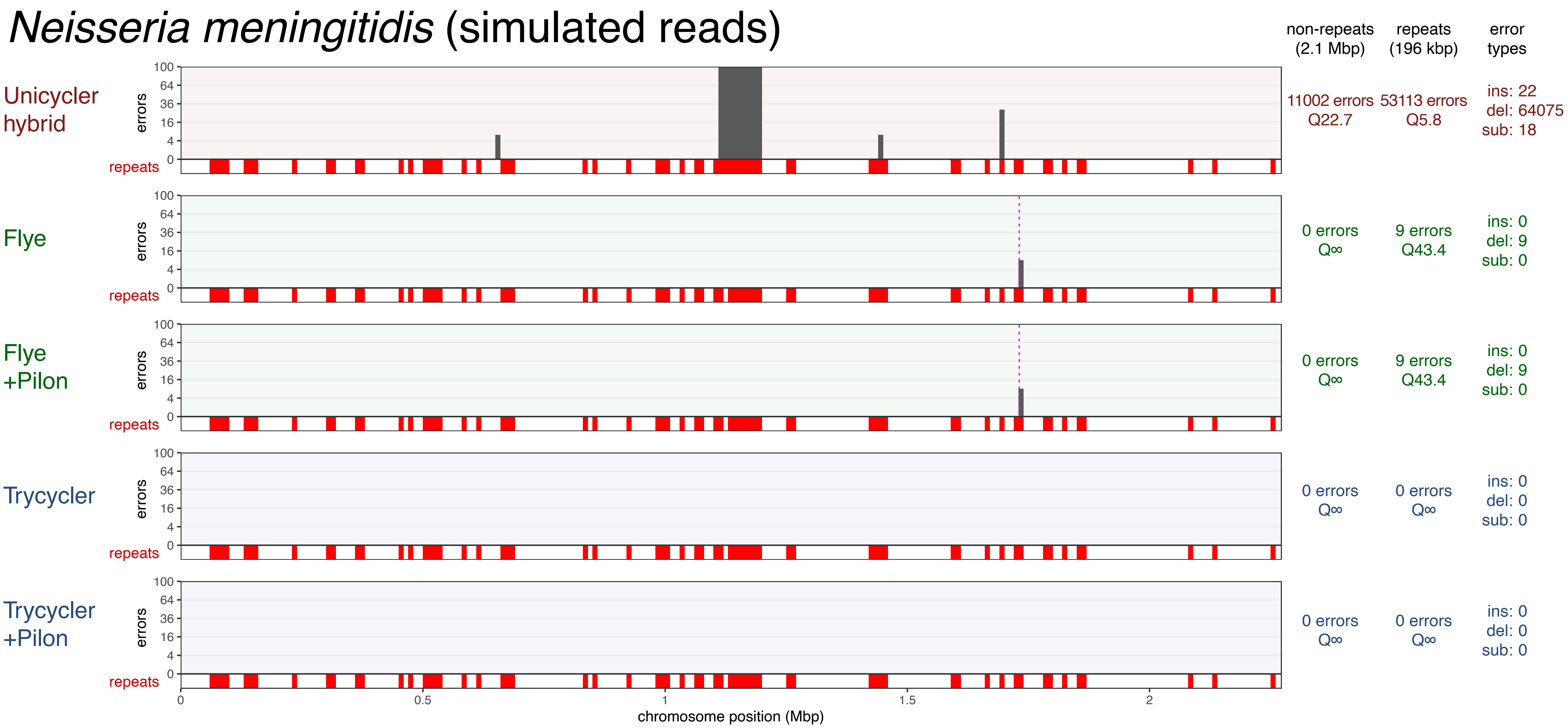


Fig. S4-f: error distributions (as defined by alignment to the reference sequence) for the simulated-read assemblies of the *Neisseria meningitidis* chromosome. Repetitive regions of the genome are shown as red blocks at the bottom of each plot. For Flye assemblies, the contig start/end position (where the sequence began before it was rotated to be consistent with the reference) is shown by the dashed line.

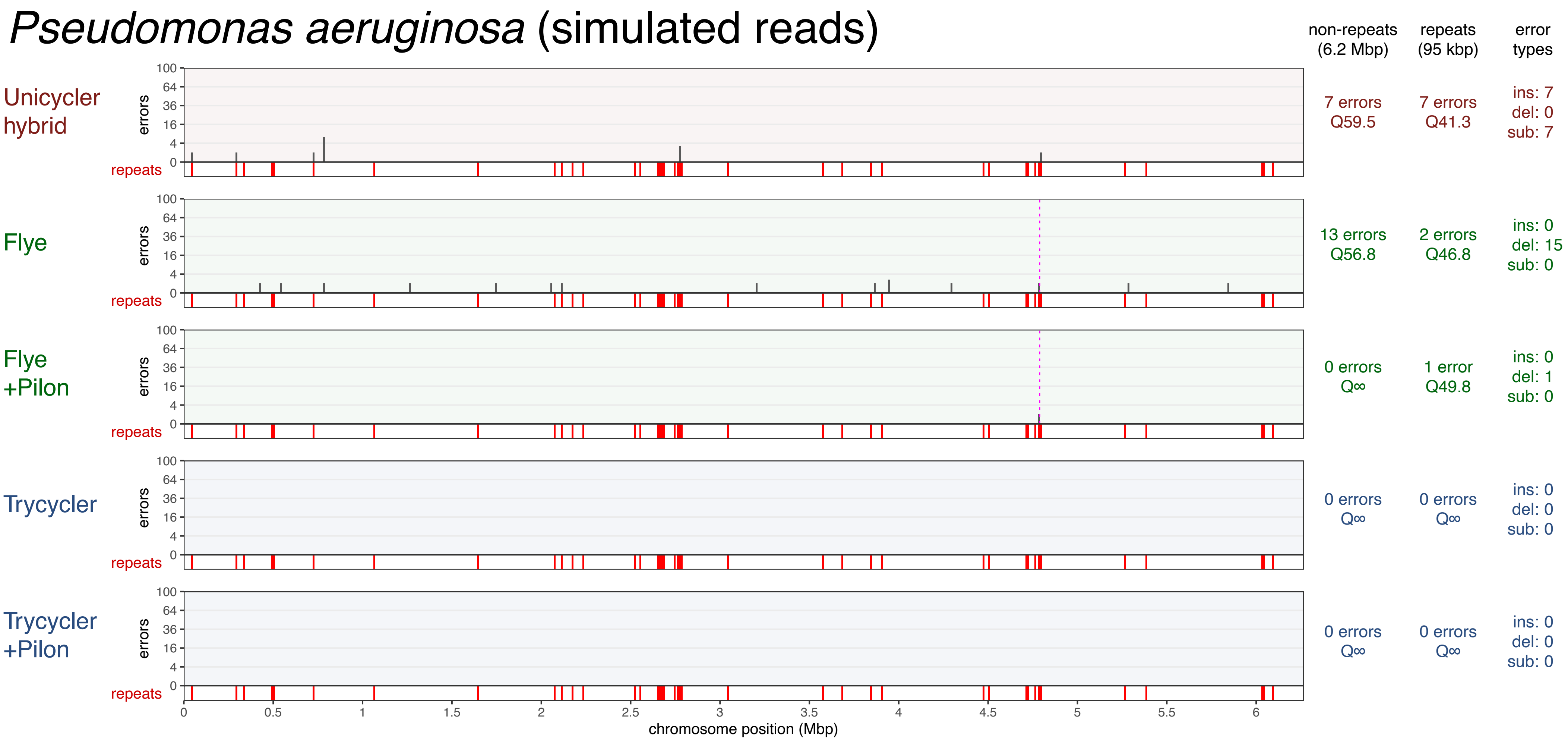
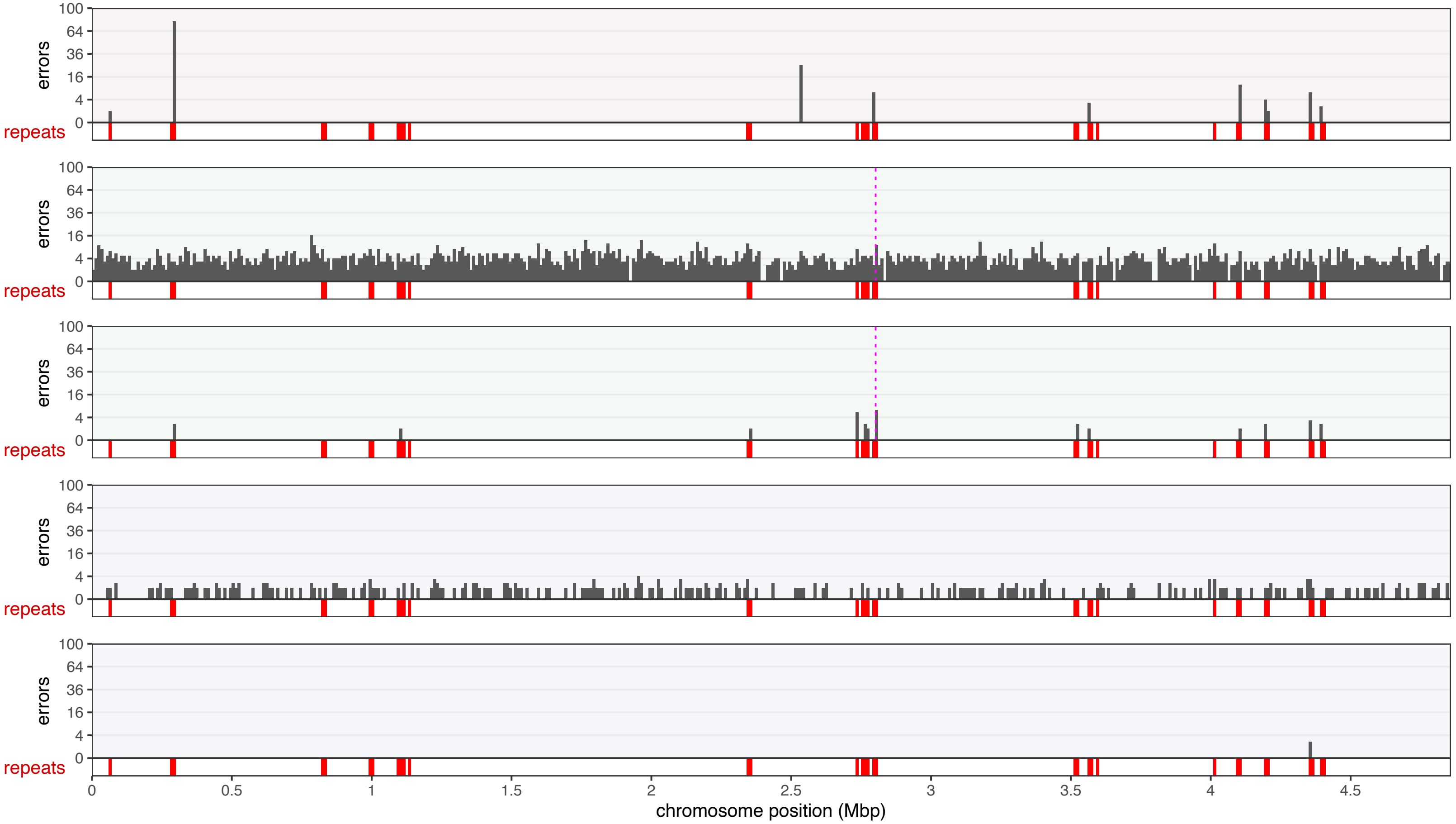


Fig. S4-g: error distributions (as defined by alignment to the reference sequence) for the simulated-read assemblies of the *Pseudomonas aeruginosa* chromosome. Repetitive regions of the genome are shown as red blocks at the bottom of each plot. For Flye assemblies, the contig start/end position (where the sequence began before it was rotated to be consistent with the reference) is shown by the dashed line.

Salmonella enterica (simulated reads)



	non-repeats (4.8 Mbp)	repeats (98 kbp)	error types
Unicycler hybrid	25 errors Q52.8	114 errors Q29.4	ins: 26 del: 28 sub: 85
Flye	1953 errors Q33.9	57 errors Q32.4	ins: 0 del: 2010 sub: 0
Flye +Pilon	6 errors Q59.0	25 errors Q36.0	ins: 0 del: 31 sub: 0
Trycycler	278 errors Q42.3	7 errors Q41.5	ins: 0 del: 285 sub: 0
Trycycler +Pilon	0 errors Q∞	2 errors Q46.9	ins: 0 del: 2 sub: 0

Fig. S4-h: error distributions (as defined by alignment to the reference sequence) for the simulated-read assemblies of the *Salmonella enterica* chromosome. Repetitive regions of the genome are shown as red blocks at the bottom of each plot. For Flye assemblies, the contig start/end position (where the sequence began before it was rotated to be consistent with the reference) is shown by the dashed line.

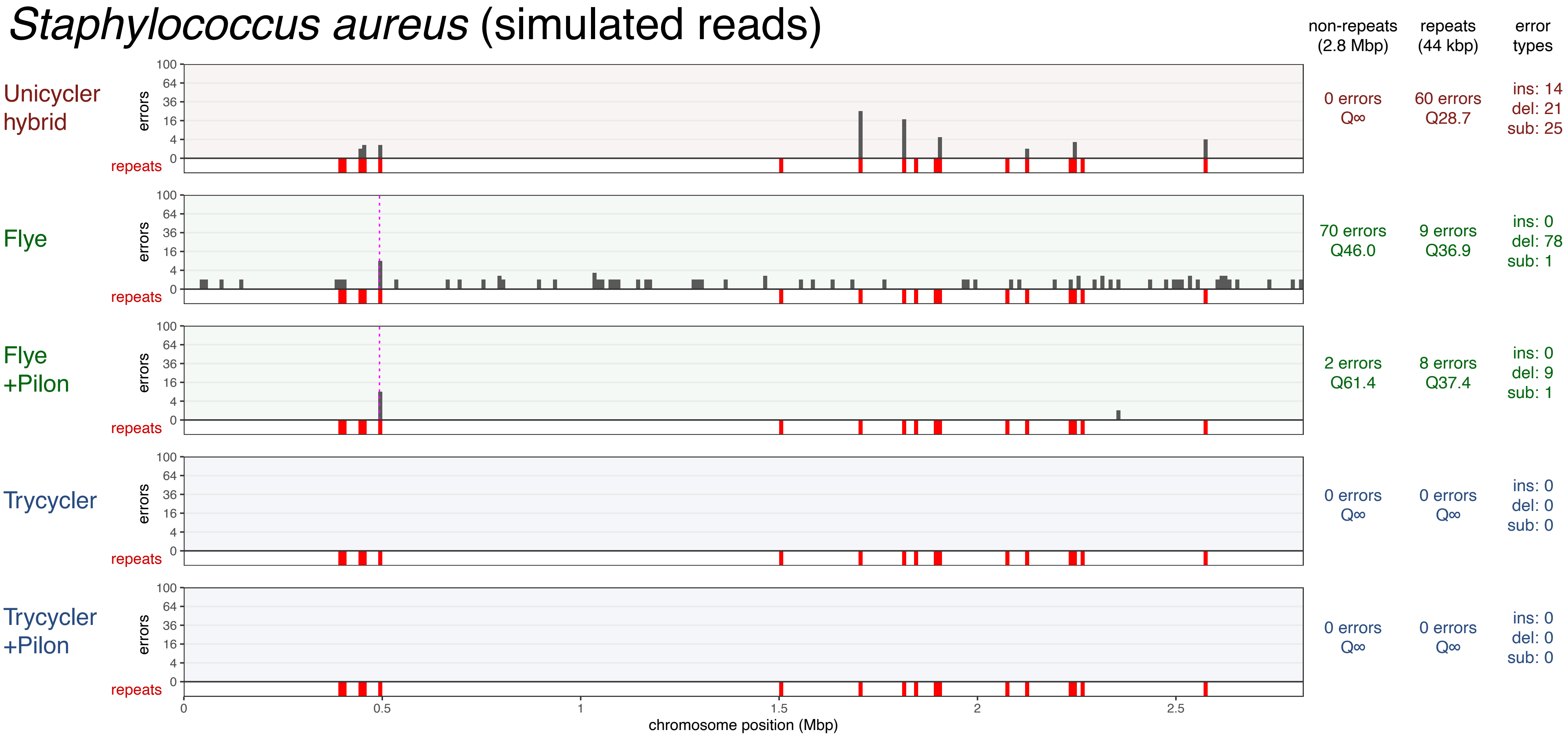


Fig. S4-i: error distributions (as defined by alignment to the reference sequence) for the simulated-read assemblies of the *Staphylococcus aureus* chromosome. Repetitive regions of the genome are shown as red blocks at the bottom of each plot. For Flye assemblies, the contig start/end position (where the sequence began before it was rotated to be consistent with the reference) is shown by the dashed line.

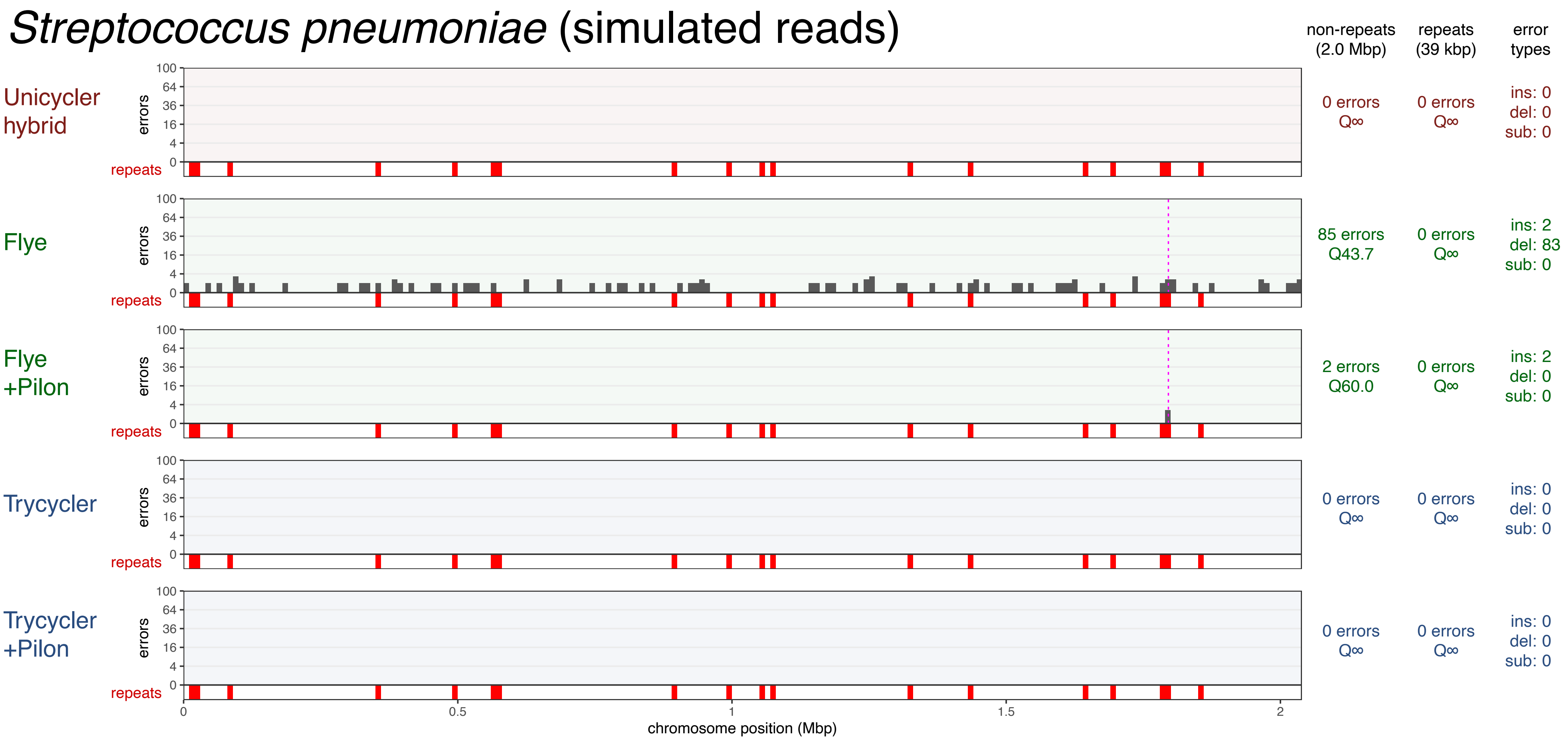


Fig. S4-j: error distributions (as defined by alignment to the reference sequence) for the simulated-read assemblies of the *Streptococcus pneumoniae* chromosome. Repetitive regions of the genome are shown as red blocks at the bottom of each plot. For Flye assemblies, the contig start/end position (where the sequence began before it was rotated to be consistent with the reference) is shown by the dashed line.

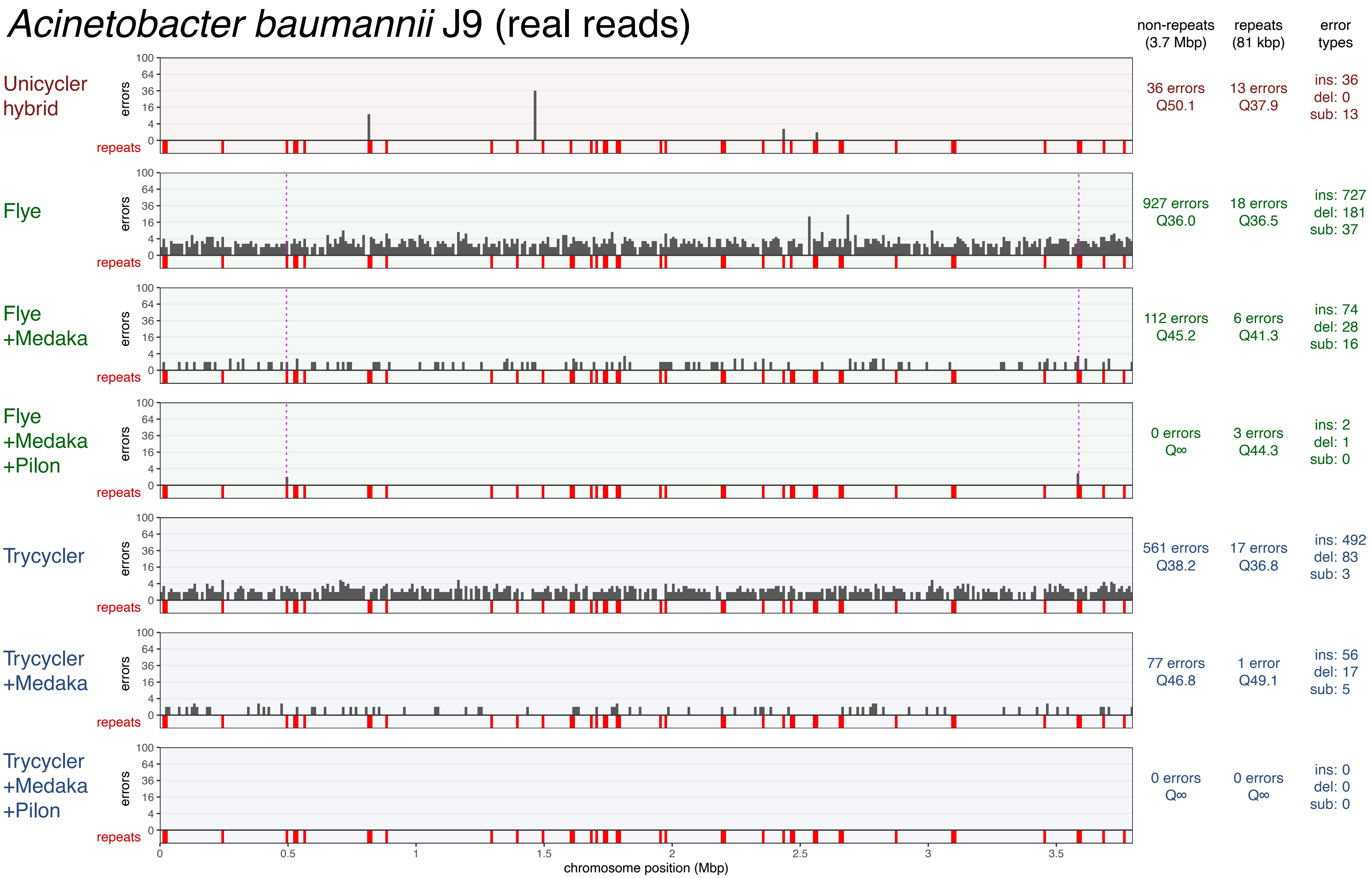


Fig. S4-k: error distributions (as defined by alignment between two independent assemblies) for the real-read assemblies of the *Acinetobacter baumannii* chromosome. Repetitive regions of the genome are shown as red blocks at the bottom of each plot. For Flye assemblies, the contig start/end positions (where the sequences began before they were rotated to be consistent with the reference) are shown by the dashed lines.

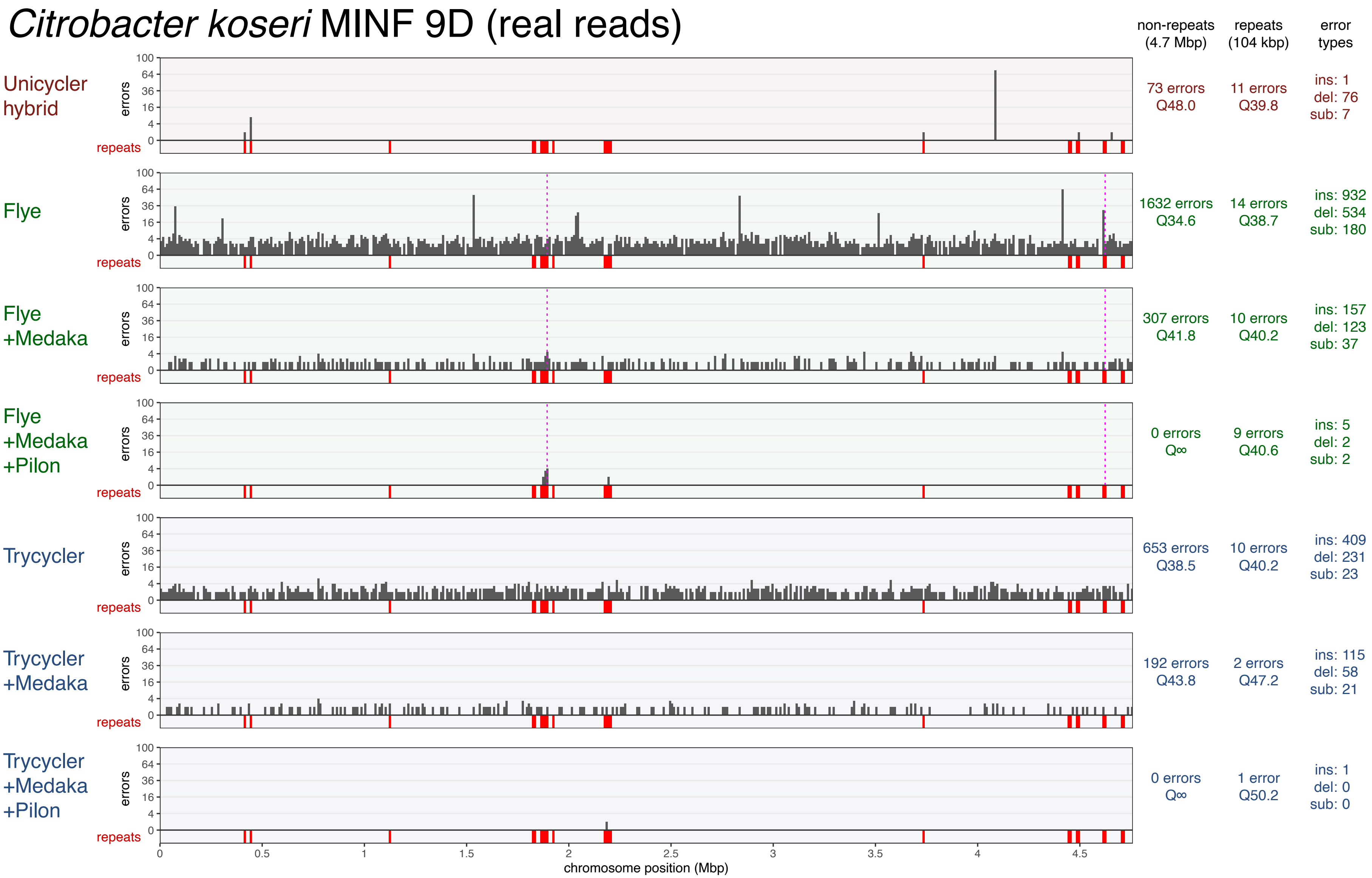
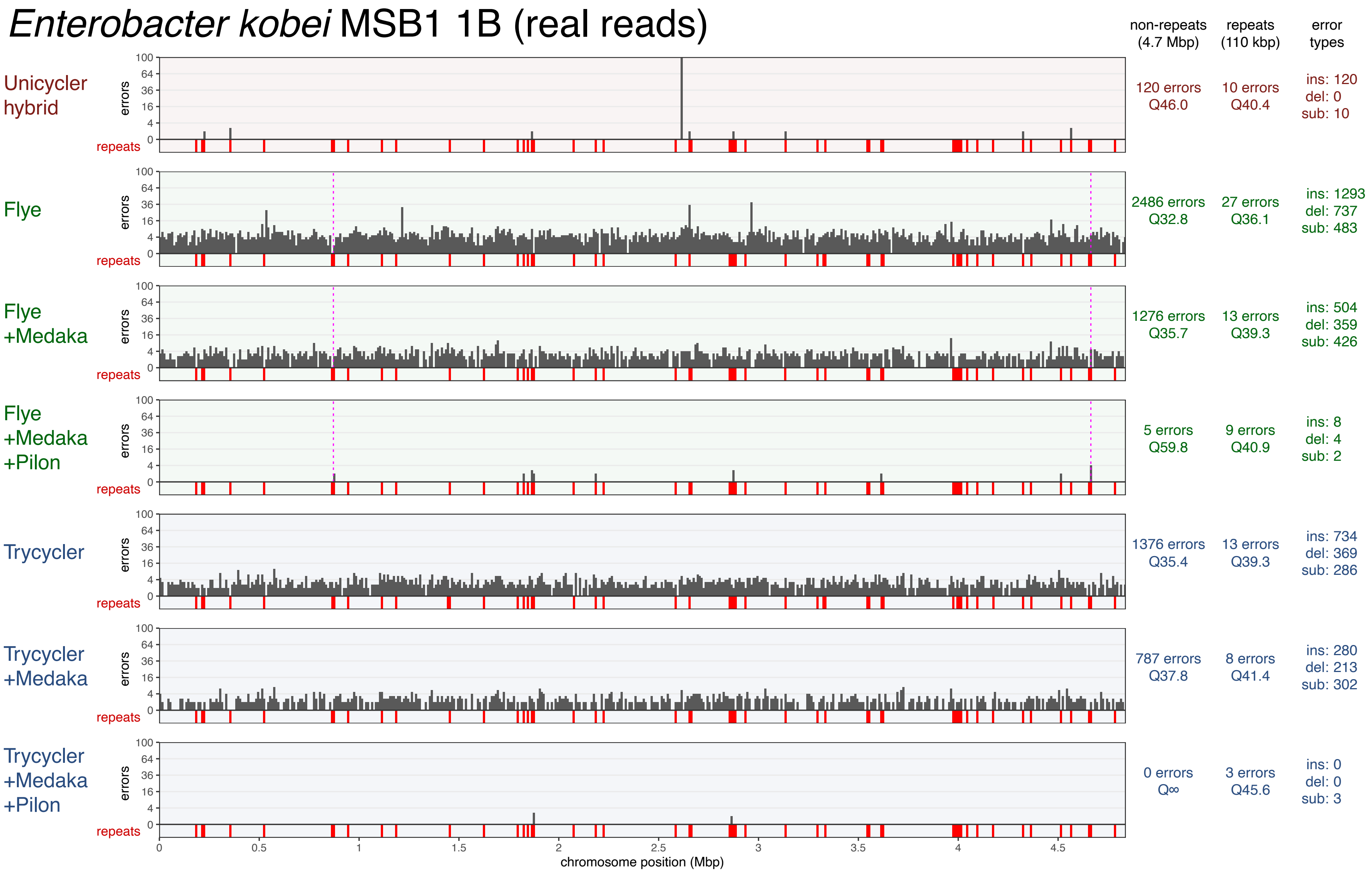


Fig. S4-l: error distributions (as defined by alignment between two independent assemblies) for the real-read assemblies of the *Citrobacter koseri* chromosome. Repetitive regions of the genome are shown as red blocks at the bottom of each plot. For Flye assemblies, the contig start/end positions (where the sequences began before they were rotated to be consistent with the reference) are shown by the dashed lines.



	non-repeats (4.7 Mbp)	repeats (110 kbp)	error types
Unicycler hybrid	120 errors Q46.0	10 errors Q40.4	ins: 120 del: 0 sub: 10
Flye	2486 errors Q32.8	27 errors Q36.1	ins: 1293 del: 737 sub: 483
Flye +Medaka	1276 errors Q35.7	13 errors Q39.3	ins: 504 del: 359 sub: 426
Flye +Medaka +Pilon	5 errors Q59.8	9 errors Q40.9	ins: 8 del: 4 sub: 2
Trycycler	1376 errors Q35.4	13 errors Q39.3	ins: 734 del: 369 sub: 286
Trycycler +Medaka	787 errors Q37.8	8 errors Q41.4	ins: 280 del: 213 sub: 302
Trycycler +Medaka +Pilon	0 errors Q∞	3 errors Q45.6	ins: 0 del: 0 sub: 3

Fig. S4-m: error distributions (as defined by alignment between two independent assemblies) for the real-read assemblies of the *Enterobacter kobei* chromosome. Repetitive regions of the genome are shown as red blocks at the bottom of each plot. For Flye assemblies, the contig start/end positions (where the sequences began before they were rotated to be consistent with the reference) are shown by the dashed lines.

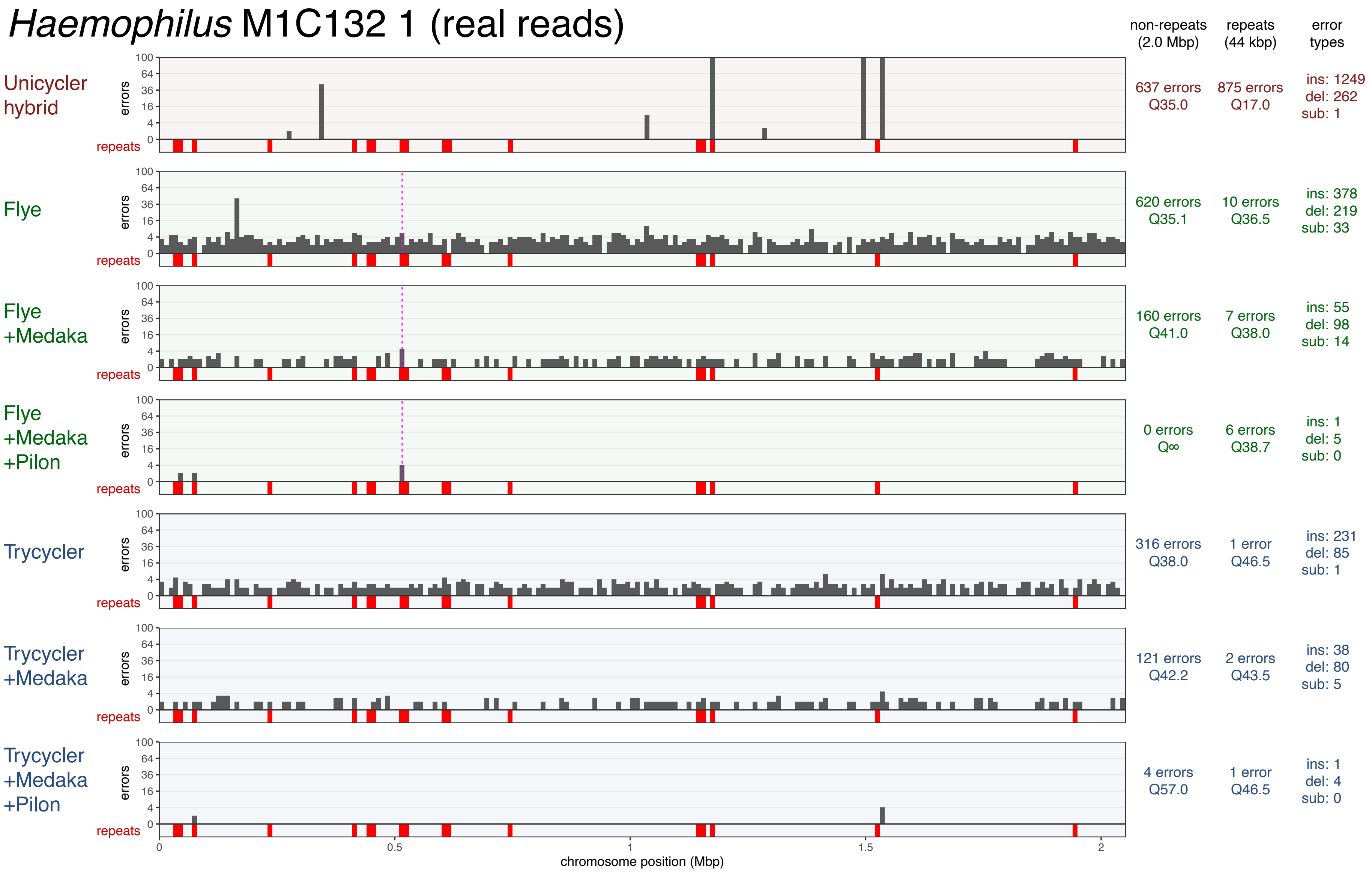


Fig. S4-n: error distributions (as defined by alignment between two independent assemblies) for the real-read assemblies of the *Haemophilus* chromosome. Repetitive regions of the genome are shown as red blocks at the bottom of each plot. For Flye assemblies, the contig start/end positions (where the sequences began before they were rotated to be consistent with the reference) are shown by the dashed lines.

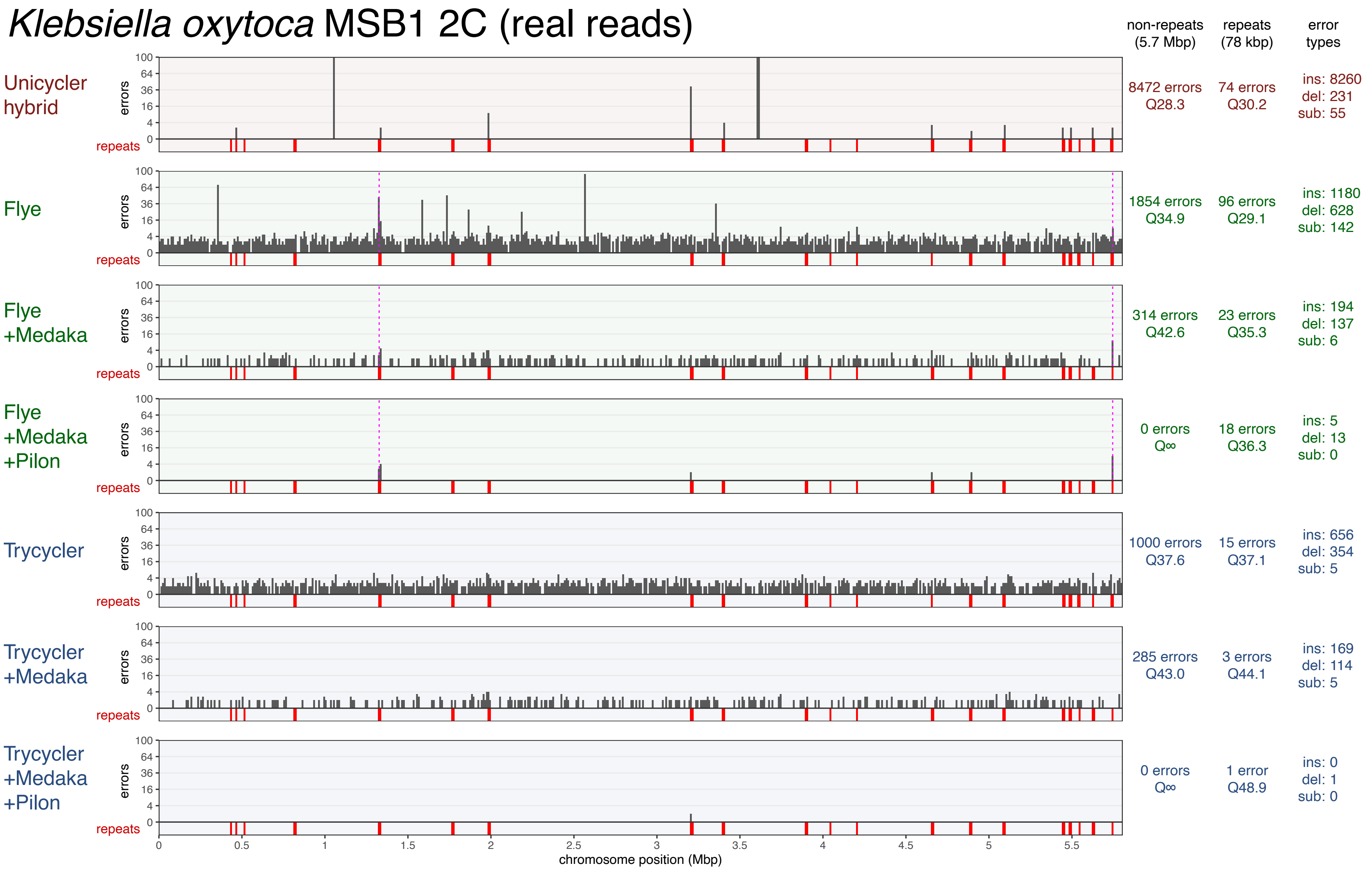


Fig. S4-o: error distributions (as defined by alignment between two independent assemblies) for the real-read assemblies of the *Klebsiella oxytoca* chromosome. Repetitive regions of the genome are shown as red blocks at the bottom of each plot. For Flye assemblies, the contig start/end positions (where the sequences began before they were rotated to be consistent with the reference) are shown by the dashed lines.

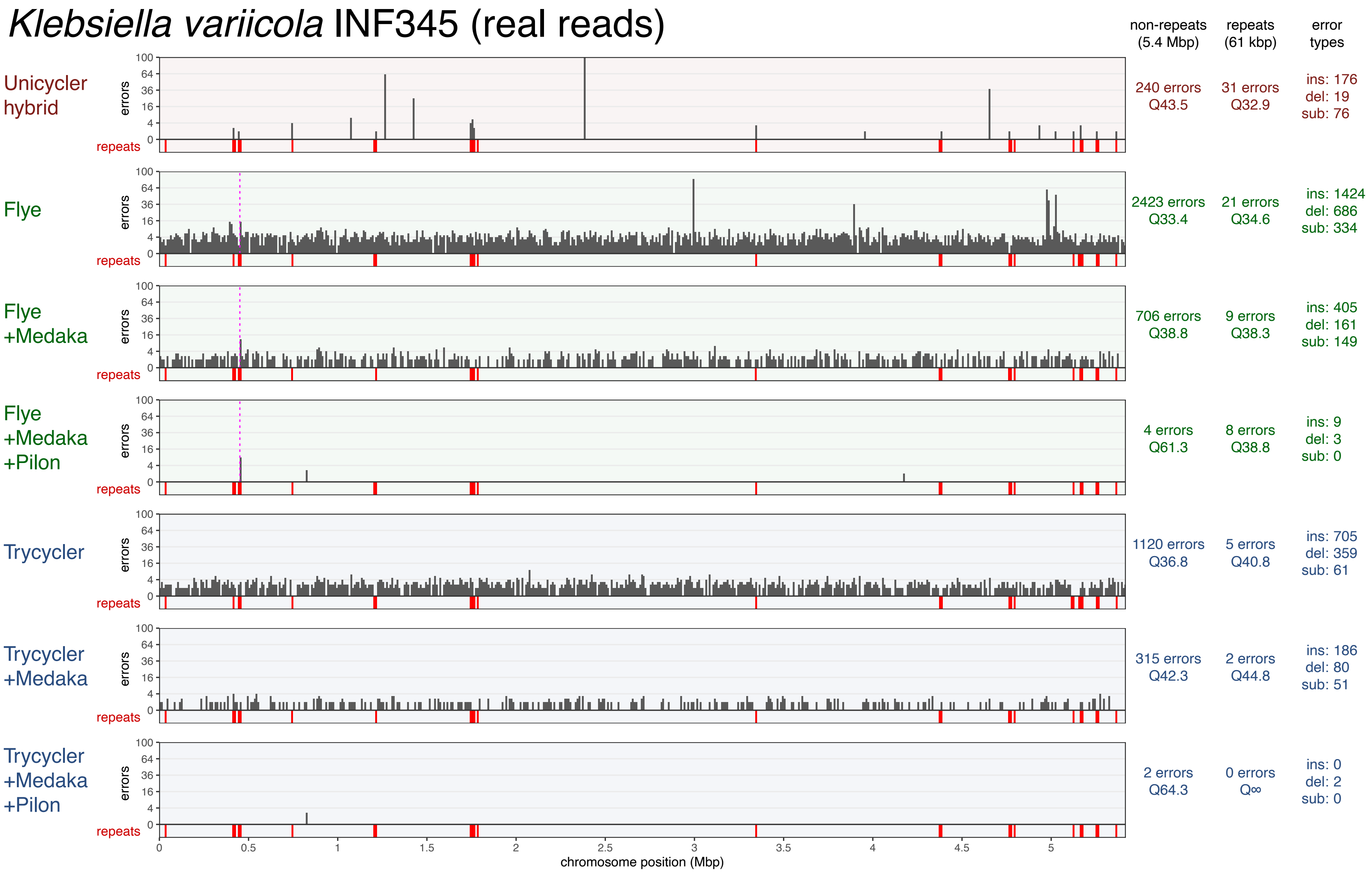


Fig. S4-p: error distributions (as defined by alignment between two independent assemblies) for the real-read assemblies of the *Klebsiella variicola* chromosome. Repetitive regions of the genome are shown as red blocks at the bottom of each plot. For Flye assemblies, the contig start/end positions (where the sequences began before they were rotated to be consistent with the reference) are shown by the dashed lines.

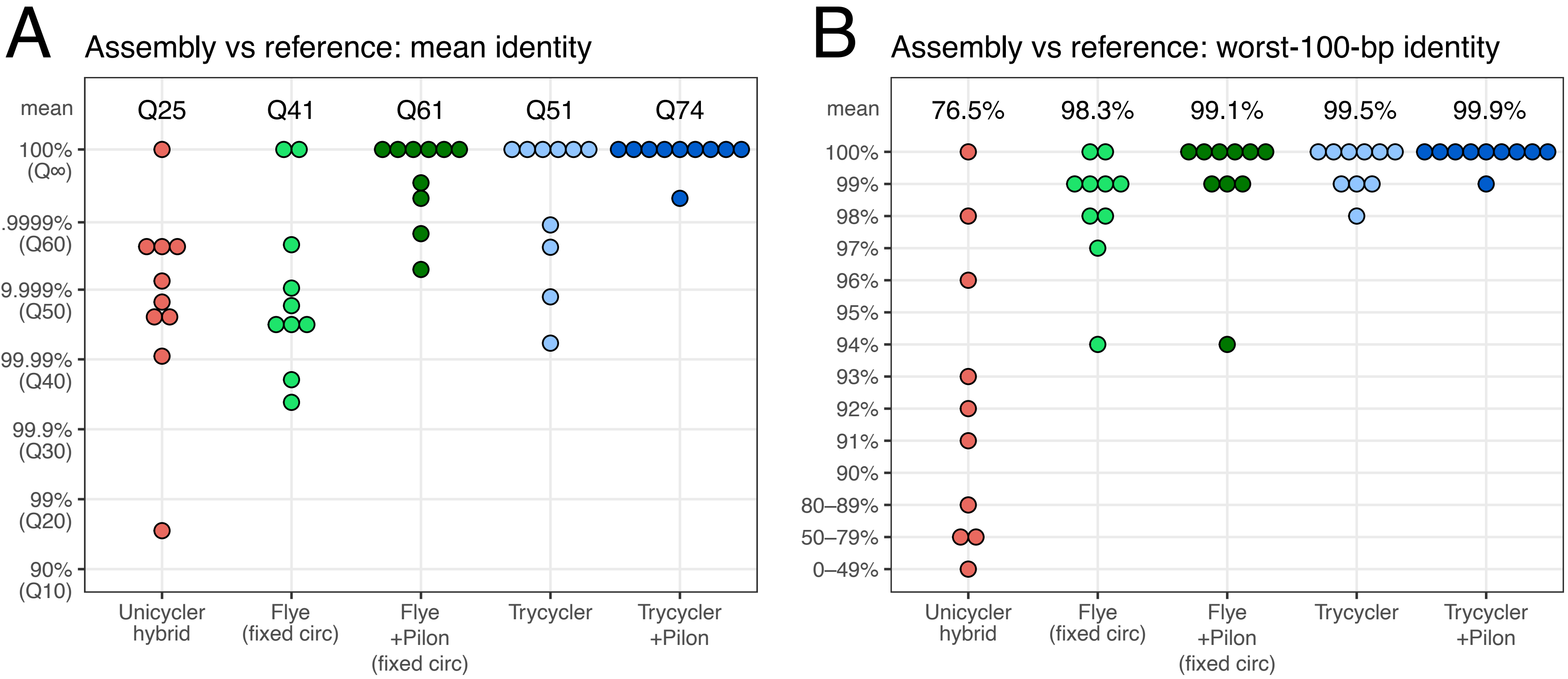


Fig. S5: results for the simulated read tests. This figure contains the same analyses as are shown in Figure 2, but the Flye assemblies have had their circularisation manually repaired.

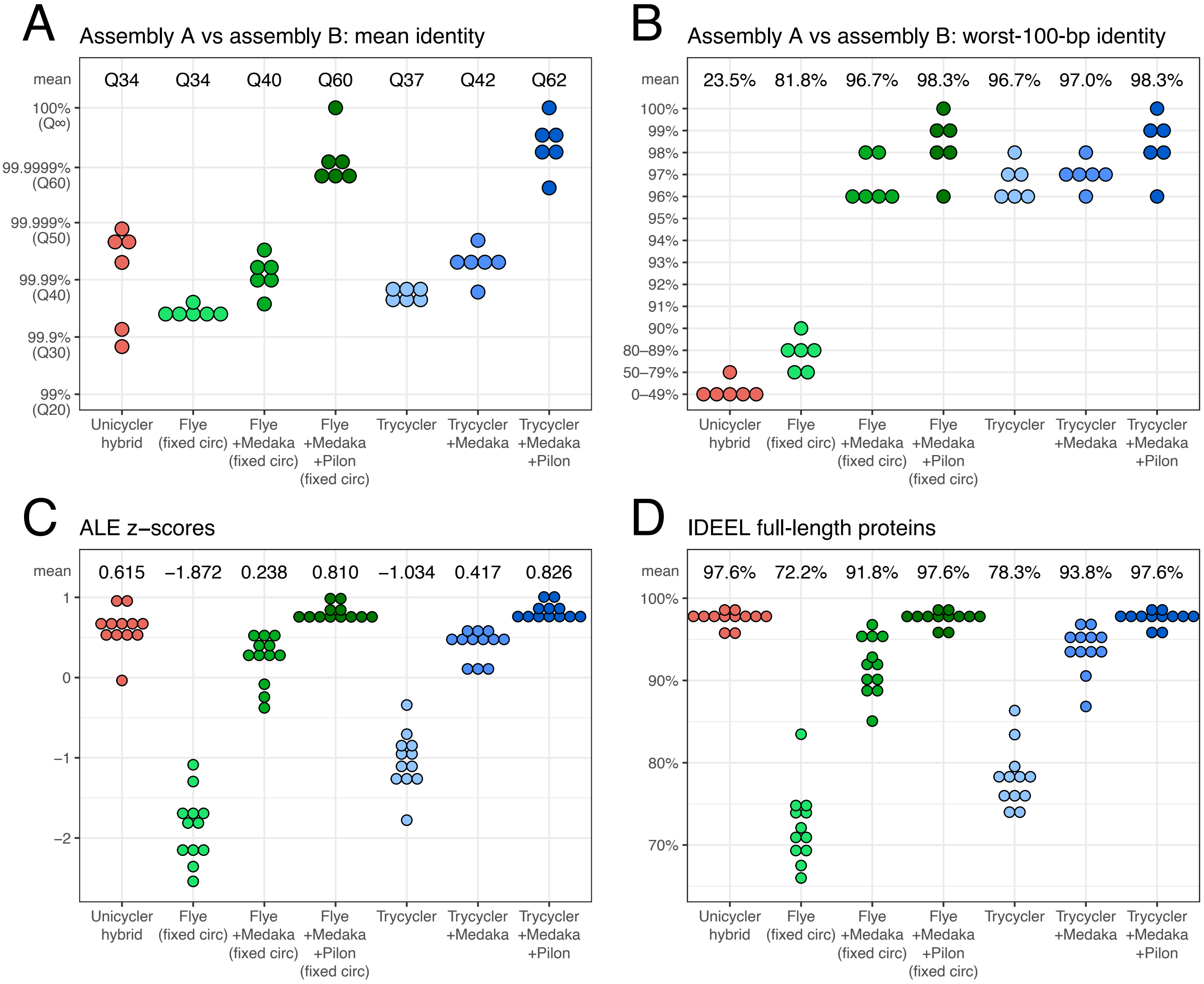


Fig. S6: results for the real read tests. This figure contains the same analyses as are shown in Figure 3, but the Flye assemblies have had their circularisation manually repaired.

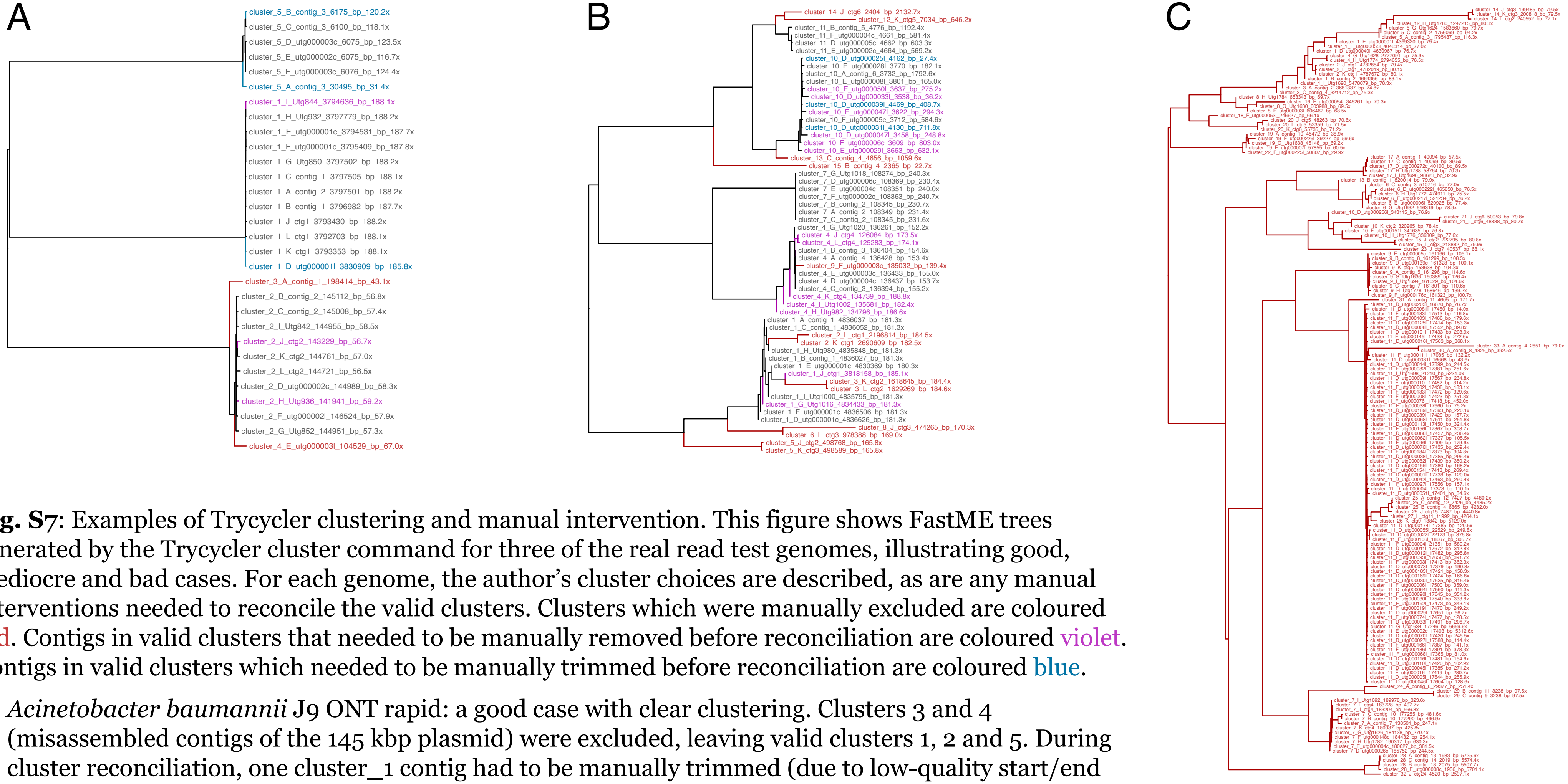

A
cluster_5_B_contig_3_6175_bp_120.2x
cluster_5_C_contig_3_6100_bp_118.1x
cluster_5_D_utg000003c_6075_bp_123.5x
cluster_5_E_utg000002c_6075_bp_116.7x
cluster_5_F_utg000003c_6076_bp_124.4x
cluster_5_A_contig_3_30495_bp_31.4x
cluster_1_I_Utg844_3794636_bp_188.1x
cluster_1_H_Utg932_3797779_bp_188.2x
cluster_1_E_utg000001c_3794531_bp_187.7x
cluster_1_F_utg000001c_3795409_bp_187.8x
cluster_1_G_Utg850_3797502_bp_188.2x
cluster_1_C_contig_1_3797505_bp_188.1x
cluster_1_A_contig_2_3797501_bp_188.2x
cluster_1_B_contig_1_3796982_bp_187.7x
cluster_1_J_ctg1_3793430_bp_188.2x
cluster_1_L_ctg1_3792703_bp_188.1x
cluster_1_K_ctg1_3793353_bp_188.1x
cluster_1_D_utg000001l_3830909_bp_185.8x
cluster_3_A_contig_1_198414_bp_43.1x
cluster_2_B_contig_2_145112_bp_56.8x
cluster_2_C_contig_2_145008_bp_57.4x
cluster_2_I_Utg842_144955_bp_58.5x
cluster_2_J_ctg2_143229_bp_56.7x
cluster_2_K_ctg2_144761_bp_57.0x
cluster_2_L_ctg2_144721_bp_56.5x
cluster_2_D_utg000002c_144989_bp_58.3x
cluster_2_H_Utg936_141941_bp_59.2x
cluster_2_F_utg000002l_146524_bp_57.9x
cluster_2_G_Utg852_144951_bp_57.3x
cluster_4_E_utg000003l_104529_bp_67.0x
B
cluster_14_J_ctg6_2404_bp_2132.7x
cluster_12_K_ctg5_7034_bp_646.2x
cluster_11_B_contig_5_4776_bp_1192.4x
cluster_11_F_utg000004c_4661_bp_581.4x
cluster_11_D_utg000005c_4662_bp_603.3x
cluster_11_E_utg000002c_4664_bp_569.2x
cluster_10_D_utg000025l_4162_bp_27.4x
cluster_10_E_utg000028l_3770_bp_182.1x
cluster_10_A_contig_6_3732_bp_1792.6x
cluster_10_E_utg000008l_3801_bp_165.0x
cluster_10_E_utg000050l_3637_bp_275.2x
cluster_10_D_utg000033l_3538_bp_36.2x
cluster_10_D_utg000039l_4469_bp_408.7x
cluster_10_E_utg000047l_3622_bp_294.3x
cluster_10_F_utg000005c_3712_bp_584.6x
cluster_10_D_utg000031l_4130_bp_711.8x
cluster_10_D_utg000047l_3458_bp_248.8x
cluster_10_F_utg000006c_3609_bp_803.0x
cluster_10_E_utg000029l_3663_bp_632.1x
cluster_13_C_contig_4_4656_bp_1059.6x
cluster_15_B_contig_4_2365_bp_22.7x
cluster_7_G_Utg1018_108274_bp_240.3x
cluster_7_D_utg000006c_108369_bp_230.4x
cluster_7_E_utg000004c_108351_bp_240.0x
cluster_7_F_utg000002c_108363_bp_240.7x
cluster_7_B_contig_2_108345_bp_230.7x
cluster_7_A_contig_2_108349_bp_231.4x
cluster_7_C_contig_2_108345_bp_231.6x
cluster_4_G_Utg1020_136261_bp_152.2x
cluster_4_J_ctg4_126084_bp_173.5x
cluster_4_L_ctg4_125283_bp_174.1x
cluster_4_B_contig_3_136404_bp_154.6x
cluster_4_A_contig_4_136428_bp_153.4x
cluster_9_F_utg000003c_135032_bp_139.4x
cluster_4_E_utg000003c_136433_bp_155.0x
cluster_4_D_utg000004c_136437_bp_153.7x
cluster_4_C_contig_3_136394_bp_155.2x
cluster_4_K_ctg4_134739_bp_188.8x
cluster_4_I_Utg1002_135681_bp_182.4x
cluster_4_H_Utg982_134796_bp_186.6x
cluster_1_A_contig_1_4836037_bp_181.3x
cluster_1_C_contig_1_4836052_bp_181.3x
cluster_2_L_ctg1_2196814_bp_184.5x
cluster_2_K_ctg1_2690609_bp_182.5x
cluster_1_H_Utg980_4835848_bp_181.3x
cluster_1_B_contig_1_4836027_bp_181.3x
cluster_1_E_utg000001c_4830369_bp_180.3x
cluster_1_J_ctg1_3818158_bp_185.1x
cluster_3_K_ctg2_1618645_bp_184.4x
cluster_3_L_ctg2_1629269_bp_184.6x
cluster_1_I_Utg1000_4835795_bp_181.3x
cluster_1_G_Utg1016_4834433_bp_181.3x
cluster_1_F_utg000001c_4836506_bp_181.3x
cluster_1_D_utg000001c_4836626_bp_181.3x
cluster_8_J_ctg3_474265_bp_170.3x
cluster_6_L_ctg3_978388_bp_169.0x
cluster_5_J_ctg2_498768_bp_165.8x
cluster_5_K_ctg3_498589_bp_165.8x
C
cluster_14_J_ctg3_199485_bp_79.5x
cluster_14_K_ctg3_200818_bp_79.5x
cluster_14_L_ctg2_240552_bp_77.1x
cluster_12_H_Utg1780_1247215_bp_80.3x
cluster_5_G_Utg1624_1583660_bp_79.7x
cluster_5_C_contig_2_1756069_bp_94.2x
cluster_5_A_contig_3_1795487_bp_116.3x
cluster_1_E_utg000001l_4369320_bp_79.4x
cluster_1_F_utg000055l_4046314_bp_77.0x
cluster_1_D_utg000049l_4630967_bp_76.7x
cluster_4_G_Utg1628_2777091_bp_75.9x
cluster_4_H_Utg1774_2794655_bp_76.5x
cluster_2_J_ctg1_4782854_bp_79.4x
cluster_2_L_ctg1_4782019_bp_80.1x
cluster_2_K_ctg1_4787672_bp_80.1x
cluster_1_B_contig_2_4664356_bp_83.1x
cluster_1_I_Utg1690_5478079_bp_78.3x
cluster_3_A_contig_2_3681337_bp_74.8x
cluster_3_C_contig_4_3214712_bp_75.3x
cluster_8_H_Utg1784_653343_bp_69.7x
cluster_16_F_utg000054l_345261_bp_70.3x
cluster_8_G_Utg1630_603988_bp_69.5x
cluster_8_E_utg000003l_606462_bp_68.5x
cluster_18_F_utg000053l_246627_bp_66.1x
cluster_20_J_ctg5_48263_bp_70.6x
cluster_20_L_ctg5_52359_bp_71.5x
cluster_20_K_ctg6_55735_bp_71.2x
cluster_19_A_contig_10_45472_bp_38.9x
cluster_19_F_utg000226l_39227_bp_59.6x
cluster_19_G_Utg1638_45148_bp_69.2x
cluster_19_E_utg000007l_57855_bp_60.5x
cluster_22_F_utg000225l_50807_bp_29.9x
cluster_17_A_contig_1_40094_bp_57.5x
cluster_17_C_contig_1_40099_bp_39.5x
cluster_17_D_utg000272c_40100_bp_89.5x
cluster_17_H_Utg1788_58764_bp_70.3x
cluster_17_I_Utg1696_98623_bp_32.9x
cluster_13_B_contig_1_820014_bp_79.9x
cluster_6_C_contig_3_510716_bp_77.0x
cluster_6_D_utg000222l_465850_bp_76.5x
cluster_6_H_Utg1772_474911_bp_75.5x
cluster_6_F_utg000217l_521234_bp_76.2x
cluster_6_E_utg000006l_520925_bp_77.4x
cluster_6_G_Utg1632_516319_bp_78.9x
cluster_10_D_utg000256l_343115_bp_76.9x
cluster_21_J_ctg6_50053_bp_79.8x
cluster_21_L_ctg6_48888_bp_80.7x
cluster_10_K_ctg2_320265_bp_78.4x
cluster_10_F_utg000151l_341635_bp_76.8x
cluster_10_H_Utg1776_336309_bp_77.6x
cluster_15_J_ctg2_222795_bp_80.8x
cluster_15_L_ctg3_218882_bp_79.9x
cluster_23_J_ctg7_40537_bp_68.1x
cluster_9_E_utg000005c_161166_bp_105.1x
cluster_9_B_contig_8_161299_bp_108.3x
cluster_9_D_utg000139c_161328_bp_100.1x
cluster_9_K_ctg5_153638_bp_104.8x
cluster_9_A_contig_5_161296_bp_114.6x
cluster_9_G_Utg1636_160389_bp_126.4x
cluster_9_I_Utg1694_161029_bp_104.6x
cluster_9_C_contig_7_161301_bp_110.6x
cluster_9_H_Utg1778_158646_bp_139.2x
cluster_9_F_utg000176c_161323_bp_100.7x
cluster_31_A_contig_11_4605_bp_171.7x
cluster_11_D_utg000203l_16670_bp_76.7x
cluster_11_D_utg000081l_17450_bp_14.0x
cluster_11_F_utg000183l_17513_bp_116.8x
cluster_11_F_utg000103l_17466_bp_179.6x
cluster_11_D_utg000125l_17414_bp_153.3x
cluster_11_D_utg000008l_17552_bp_39.8x
cluster_11_D_utg000101l_17433_bp_203.9x
cluster_11_F_utg000145l_17433_bp_272.6x
cluster_11_D_utg000016l_17563_bp_368.1x
cluster_33_A_contig_4_2651_bp_79.0x
cluster_30_A_contig_8_4825_bp_392.5x
cluster_11_F_utg000111l_17085_bp_132.2x
cluster_11_D_utg000031l_16668_bp_43.6x
cluster_11_D_utg000014l_17899_bp_244.5x
cluster_11_F_utg000082l_17381_bp_251.6x
cluster_11_I_Utg1698_21210_bp_5231.0x
cluster_11_D_utg000009l_17667_bp_234.8x
cluster_11_F_utg000010l_17482_bp_314.2x
cluster_11_F_utg000002l_17438_bp_183.1x
cluster_11_F_utg000133l_17472_bp_329.6x
cluster_11_F_utg000008l_17423_bp_251.3x
cluster_11_F_utg000076l_17418_bp_452.0x
cluster_11_F_utg000038l_17660_bp_75.2x
cluster_11_D_utg000189l_17393_bp_220.1x
cluster_11_F_utg000039l_17429_bp_157.7x
cluster_11_D_utg000068l_17511_bp_251.8x
cluster_11_D_utg000113l_17450_bp_321.4x
cluster_11_F_utg000156l_17367_bp_308.7x
cluster_11_D_utg000066l_17437_bp_236.4x
cluster_11_D_utg000062l_17337_bp_105.5x
cluster_11_F_utg000096l_17409_bp_179.6x
cluster_11_D_utg000076l_17435_bp_259.4x
cluster_11_F_utg000184l_17373_bp_304.8x
cluster_11_D_utg000038l_17385_bp_296.4x
cluster_11_D_utg000082l_17439_bp_350.2x
cluster_11_D_utg000155l_17380_bp_168.2x
cluster_11_F_utg000154l_17413_bp_269.4x
cluster_11_D_utg000001l_17738_bp_120.0x
cluster_11_D_utg000042l_17463_bp_290.4x
cluster_11_F_utg000027l_17556_bp_157.1x
cluster_11_D_utg000004l_17373_bp_110.1x
cluster_11_D_utg000051l_17401_bp_34.6x
cluster_25_A_contig_12_7427_bp_4480.2x
cluster_25_C_contig_12_7426_bp_4485.2x
cluster_25_B_contig_4_6865_bp_4282.0x
cluster_25_J_ctg15_7487_bp_4440.8x
cluster_27_L_ctg11_11992_bp_4264.1x
cluster_26_K_ctg9_13842_bp_5129.0x
cluster_11_D_utg000174l_17385_bp_120.5x
cluster_11_D_utg000055l_22529_bp_249.8x
cluster_11_D_utg000022l_22123_bp_376.8x
cluster_11_F_utg000106l_18667_bp_305.7x
cluster_11_F_utg000004l_21351_bp_580.2x
cluster_11_D_utg000011l_17672_bp_312.8x
cluster_11_D_utg000012l_17482_bp_295.8x
cluster_11_F_utg000093l_17656_bp_391.7x
cluster_11_F_utg000003l_17413_bp_362.3x
cluster_11_D_utg000073l_17379_bp_190.8x
cluster_11_D_utg000183l_17421_bp_158.3x
cluster_11_D_utg000169l_17424_bp_166.8x
cluster_11_D_utg000030l_17535_bp_315.4x
cluster_11_F_utg000006l_17500_bp_359.0x
cluster_11_D_utg000064l_17560_bp_411.3x
cluster_11_F_utg000090l_17645_bp_351.2x
cluster_11_F_utg000030l_17540_bp_333.8x
cluster_11_F_utg000192l_17473_bp_343.1x
cluster_11_F_utg000019l_17470_bp_249.2x
cluster_11_D_utg000029l_17651_bp_56.7x
cluster_11_F_utg000074l_17477_bp_128.5x
cluster_11_D_utg000033l_17491_bp_206.7x
cluster_11_G_Utg1634_17246_bp_6659.6x
cluster_11_E_utg000002c_17403_bp_5312.6x
cluster_11_D_utg000070l_17430_bp_245.5x
cluster_11_D_utg000027l_17588_bp_114.4x
cluster_11_F_utg000166l_17387_bp_141.1x
cluster_11_F_utg000186l_17391_bp_378.3x
cluster_11_F_utg000068l_17365_bp_81.0x
cluster_11_D_utg000116l_17481_bp_154.6x
cluster_11_D_utg000110l_17420_bp_102.9x
cluster_11_D_utg000045l_17385_bp_271.2x
cluster_11_F_utg000016l_17419_bp_280.7x
cluster_11_D_utg000005l_17644_bp_255.9x
cluster_11_D_utg000046l_17604_bp_128.6x
cluster_24_A_contig_6_29377_bp_251.4x
cluster_29_B_contig_11_3238_bp_97.5x
cluster_29_C_contig_9_3238_bp_97.5x
cluster_7_I_Utg1692_189978_bp_323.6x
cluster_7_L_ctg4_183728_bp_497.7x
cluster_7_J_ctg4_183204_bp_566.8x
cluster_7_C_contig_10_177255_bp_481.6x
cluster_7_B_contig_10_177290_bp_466.9x
cluster_7_A_contig_7_138501_bp_247.1x
cluster_7_K_ctg4_180037_bp_425.8x
cluster_7_G_Utg1626_184138_bp_270.4x
cluster_7_F_utg000148c_184432_bp_254.1x
cluster_7_H_Utg1782_190317_bp_630.3x
cluster_7_E_utg000004c_180627_bp_381.5x
cluster_7_D_utg000026c_185752_bp_244.5x
cluster_28_A_contig_13_1983_bp_5725.6x
cluster_28_C_contig_14_2019_bp_5574.4x
cluster_28_B_contig_13_2075_bp_5507.7x
cluster_28_E_utg000008c_1936_bp_5701.1x
cluster_32_J_ctg24_4520_bp_2597.1x

Fig. S7: Examples of Trycycler clustering and manual intervention. This figure shows FastME trees generated by the Trycycler cluster command for three of the real read test genomes, illustrating good, mediocre and bad cases. For each genome, the author's cluster choices are described, as are any manual interventions needed to reconcile the valid clusters. Clusters which were manually excluded are coloured red. Contigs in valid clusters that needed to be manually removed before reconciliation are coloured violet. Contigs in valid clusters which needed to be manually trimmed before reconciliation are coloured blue.

A. *Acinetobacter baumannii* J9 ONT rapid: a good case with clear clustering. Clusters 3 and 4 (misassembled contigs of the 145 kbp plasmid) were excluded, leaving valid clusters 1, 2 and 5. During cluster reconciliation, one cluster_1 contig had to be manually trimmed (due to low-quality start/end sequence), one cluster_1 contig had to be manually removed (due to poor pairwise alignment), two cluster_2 contigs had to be manually removed (due to being incomplete) and two cluster_5 contigs had to be manually trimmed (due to excessive length).

B. *Enterobacter kobei* MSB1_1B ONT rapid: a mediocre case with more complex clusters. Clusters 2, 3, 5, 6, 8 (misassembled contigs of the chromosome), 9 (misassembled contig of the 136 kbp plasmid), 12, 13, 14, and 15 (misassembled contigs of the small plasmids) were excluded, leaving valid clusters 1, 4, 7, 10 and 11. During cluster reconciliation, two cluster_1 contigs had to be manually removed (due to being incomplete or poor pairwise alignment), five cluster_4 contigs had to be manually removed (due to being incomplete), three cluster_10 contigs had to be manually trimmed (due to excessive length) and six cluster_10 contigs had to be manually removed (due to being incomplete or unable to circularise).

C. *Serratia marcescens* 17-147-1671 ONT rapid: a bad case where valid clusters were unclear. Insufficient read length and genome heterogeneity both contributed to the poor results. Without good clusters, it was not possible to proceed with Trycycler assembly.