

Supplementary Information (SI): Training alignment parameters for arbitrary sequencers with LAST-TRAIN

Michiaki Hamada, Yukiteru Ono, Kiyoshi Asai and Martin C. Frith

S1 Detailed methods in LAST-TRAIN

S1.1 Steps in LAST-TRAIN

LAST-TRAIN runs `lastal` to find alignments between the given query and reference sequences, with `lastal` option `-j7`, which gets the expected counts of substitutions, gap opens and extensions for each alignment. `lastal` may find more than one alignment per query, including unwanted paralogs, e.g. it may align an alpha-globin DNA read to both alpha-globin and beta-globin. Therefore, LAST-TRAIN filters the alignments through LAST-SPLIT, which discards alignments that do not include a confident unique best match for any part of the query (Frith and Kawaguchi, 2015).

S1.2 Calculation of expected counts by `lastal`

As described previously (Frith *et al.*, 2010b; Hamada *et al.*, 2011), `lastal` uses an X-drop algorithm to extend a gapped alignment to the left and right of an alignment “core”. Each X-drop extension explores a limited area of the dynamic programming matrix (Altschul *et al.*, 1997; Zhang *et al.*, 1998). `lastal` then evaluates the probabilities of alternative alignments, by performing a forward-backward algorithm (Durbin *et al.*, 1998) within the dynamic programming region defined by the preceding X-drop algorithm. It then calculates the expected substitution and gap counts, based on the probabilities of alternative alignments.

- c_{xy} : expected count of letter type x aligned to letter type y .
- c_m : expected count of matches plus mismatches.
- c_D : expected count of deleted letters.
- c_I : expected count of inserted letters.
- c_d : expected count of deletion opens (= count of deletion closes).
- c_i : expected count of insertion opens (= count of insertion closes).

Also, c_a is the count of alignments.

S1.3 Score parameters from expected counts

LAST-TRAIN infers score parameters from these expected counts, based on the alignment model in Figure S1. First, it infers the substitution probabilities (π_{xy}), the deletion open (α_D) and extension (β_D) probabilities, and the insertion open (α_I) and extension (β_I) probabilities, using these formulas:

$$\begin{aligned}\pi_{xy} &= c_{xy} / \sum_{x,y} c_{xy} \\ \alpha_D &= c_d / (c_m + c_d + c_i + c_a), \\ \alpha_I &= c_i / (c_m + c_d + c_i + c_a) \\ \beta_D &= (c_D - c_d) / c_D, \\ \beta_I &= (c_I - c_i) / c_I\end{aligned}$$

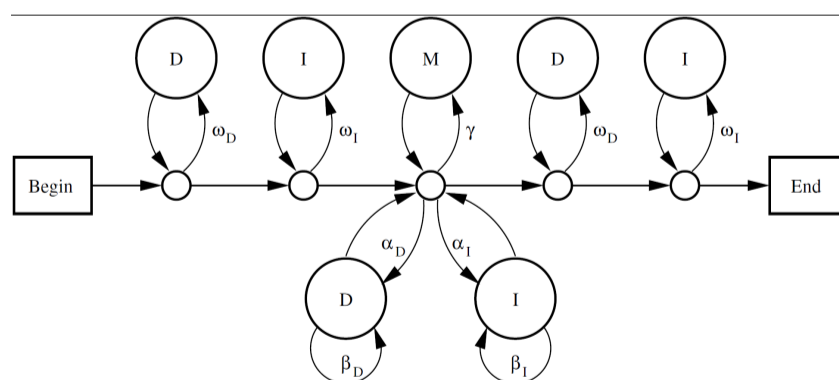


Fig. S1. A statistical model for pair-wise local sequence alignment. States labeled D (“deletion”) emit reference letters x with probability ϕ_x . States labeled I (“insertion”) emit query letters y with probability ψ_y . The state labeled M emits aligned letters $x : y$ with probability π_{xy} . The small unlabeled circles are just connectors, and do not emit. In the case of overlap alignment, either D or I can be utilized in the left- or right-endpoints.

Using the model in Figure S1, maximum-likelihood alignments are the same as maximum-score alignments with this scoring scheme (where t is an arbitrary scale factor):

$$\begin{aligned}
 S(x, y) &= t \ln \left(\frac{\pi_{xy}}{\phi_x \psi_y} \cdot \frac{\gamma}{\omega_D \omega_I} \right) && \text{substitution score matrix} \\
 a_D &= t \ln \left(\frac{\alpha_D (1 - \beta_D)}{\beta_D} \right) && \text{delete existence score} \\
 b_D &= t \ln \left(\frac{\beta_D}{\omega_D} \right) && \text{delete extension score} \\
 a_I &= t \ln \left(\frac{\alpha_I (1 - \beta_I)}{\beta_I} \right) && \text{insert existence score} \\
 b_I &= t \ln \left(\frac{\beta_I}{\omega_I} \right) && \text{insert extension score}
 \end{aligned}$$

Here, “existence score” means that a length- k gap scores: existence score + $k \times$ extension score. LAST-TRAIN gets the score parameters using these formulas, assuming that:

$$\begin{aligned}
 \phi_x &= \sum_y \pi_{xy}, \\
 \psi_y &= \sum_x \pi_{xy} \\
 \gamma / (\omega_D \omega_I) &\approx 1, \\
 \beta_D / \omega_D &\approx \beta_D, \\
 \beta_I / \omega_I &\approx \beta_I
 \end{aligned}$$

The scores are rounded to nearest integers (but if b_D or b_I would be 0, it is set to -1).

S1.4 Pseudo-counts

To avoid problems when an expected count is very small, LAST-TRAIN uses +1 pseudo-counts: $c_a \leftarrow c_a + 1$, $c_m \leftarrow c_m + 1$, $c_d \leftarrow c_d + 1$, $c_i \leftarrow c_i + 1$, $c_{xy} \leftarrow c_{xy} + 1$, $c_D \leftarrow c_D + 2$, $c_I \leftarrow c_I + 2$. (c_D and c_I use +2 because they are really opens + extensions.)

S1.5 Scale factor t

The value of t would have no effect on the results, if the scores were not rounded to integers. Since they are, t should not be too low, to avoid excessive accuracy loss from rounding. On the other hand, a very high value of t would produce scores with an unwarranted and misleading number of significant figures, and the training would take longer to converge.

LAST-TRAIN starts with an initial substitution score matrix, which can be chosen by the user (though we believe the default is usually OK). This initial matrix corresponds to some value of t , which LAST infers using the Method of Yu and Altschul (2005). LAST-TRAIN uses this same value of t to calculate the final set of parameters, i.e. the output parameters use the same scale as the initial parameters. While training, however, the scale is multiplied by 20: without this, the integer-rounded score parameters can stop changing too soon.

S1.6 Score threshold

At each iteration of training, lastal reports alignments with score \geq a threshold. This threshold is based on statistical significance. By default, it is the minimum score such that at most one alignment with greater or equal score is expected per million query bases, for random sequences with letter frequencies ϕ and ψ . It is not easy to evaluate significance for arbitrary score parameters, and this is an important enabler of training (Sheetlin et al., 2016).

S1.7 Strand asymmetry

The trained substitution score matrix may lack strand symmetry, e.g. the a:g score may not equal the t:c score, which requires careful handling. A LAST option specifies whether the matrix applies to: (i) the reference forward strand aligned to either query strand, or (ii) either reference strand aligned to the query forward strand. The latter is LAST-TRAIN’s default, which was used throughout this study.

S1.8 Overlap alignment

In this study we tested *overlap* alignment (Durbin et al., 1998), where each alignment is extended in both directions until it hits the end of one or other sequence. (When aligning shorter reads to longer chromosomes, of course the alignments will usually reach the ends of the reads rather than the chromosomes.) LAST’s implementation of this uses its normal X-drop algorithm, so if an alignment cannot reach the end of a sequence without a score drop $> x$, no alignment is reported. In mapping a read to a reference genome, overlap alignments are better than local alignments if the read is expected to be a complete subsequence of the reference genome, while local alignments are better if a read has adapter sequences, rearrangements, etc. LAST cannot evaluate significance for overlap alignment, so it uses the same score threshold as for local alignment, which is conservative.

S1.9 LAST-TRAIN options

In our experiments, we ran LAST-TRAIN with option `-T0` for local alignment, and `-T1` for overlap alignment, where the version of LAST is 658. The following initial parameters for training were used: +5/-5 match/mismatch scores and 15/3 gap open/extend costs.

S2 Datasets

In this study, we employ three types of datasets generated by PacBio, IonTorrent and Nanopore sequencers summarized in Tables S1, S2 and S5, respectively. These sequencers are categorized as 3rd or 4th generation sequencers, and have been recently utilized in many studies e.g. Loman *et al.* (2015); Chin *et al.* (2013).

S2.1 PacBio RS

We used 11 PacBio datasets freely available from the Web. Table S1 shows a summary of PacBio RS data used in this study, including various versions of the sequencing technology, where X-Y (e.g. P4-C2) means the sequencer version is Y and the chemistry version (e.g. polymerase) is X. The reads come from several reference genomes whose GC% varies from 35% (*C.elegans*) to 63% (*M.ruber*). It should be noted that the latest version of PacBio is P6-C4 as of Jul 25, 2015.

PacBio RS generates two types of reads, CLR (continuous long read; long and error-prone) and CCS (circular consensus sequencing; short and low error rate; see e.g. Ono *et al.* (2013)). In this study, we focus only on CLR, because most studies that utilize PacBio sequencers employed CLR rather than CCS, in order to receive benefit from the *long* reads. In addition, CLR includes more errors than CCS and it is expected to be useful to train alignment parameters for these erroneous reads with characteristic error profiles. See Ono *et al.* (2013) for a detailed investigation of error characteristics for PacBio reads including CLR and CCS (with older versions e.g. C2-C2).

Table S1. Summary of PacBio RS read data utilized in this study

(i) Reference	(ii) Genome size	(iii) %GC	(iv) version	(v) ave.len.	(vi) #reads	(vii) notes
E-coli K12 MG1655	4,639,675	50.79%	C2-C2	2,997	31,815	*1
			XL-C2	3,501	130,753	*2
			P4-C2	5,433	61,019	*2
V.cholerae N5	3,718,269	47.48%	C2-C2	5,690	76,129	*1
M.ruber	3,098,881	63.38%	XL-C2	2,710	121,513	*2
P.heparinus	5,167,383	42.05%	XL-C2	2,672	179,130	*2
S.cerevisiae	12,157,105	38.15%	P4-C2	6,350	107,872	*2,*3
N.creassa	41,102,378	48.19%	P4-C3	5,467	102,937	*2
H.sapiens	3,095,693,983	40.90%	P5-C3	7,561	22,081	*2,*4
D.melanogaster	120,401,063	42.38%	P5-C3	10,521	115,069	*2,*5
C.elegans	100,286,070	35.44%	P6-C4	11,974	8,261	*2,*6

Each column indicates (i) reference genome of read sequences, (ii) the genome size of the reference, (iii) GC % of the reference genome, (iv) the version of PacBio sequencing technology where the 1st part indicates the enzyme version and the 2nd part indicates the sequencer version, (v) the average read length, and (vi) the number of reads. The 7-th column (vii) shows notes as follows: (*1) The identical data was used in a previous study (Ono *et al.*, 2013), where we filtered FASTQ files with the following filtering criteria: length > 100 bp and quality > 75%. (*2) We filtered `bas.h5`, `bax.h5m`, `fastq` files with the following conditions: length > 500bp and quality > 80%; this condition is the default in HGAP (Chin *et al.*, 2013). (*3) For *S.cerevisiae*, reads were taken from `w303` although the reference genome is `S288C`. (*4) For *H.sapiens* part of data was utilized: the total number of reads is 21,856,161 and the average read length is equal to 7,680. (*5) For *D.melanogaster*, part of data was utilized: the total number of reads is 1,514,730 and the average read length is 10,040. (*6) For *C.elegans*, part of data was utilized.

S2.2 IonTorrent

In this study, we used Ion Xpress 200 data reported in Ross *et al.* (2013). Figure S2 shows the read length distribution. For quality control, FASTXToolkit (http://hannonlab.cshl.edu/fastx_toolkit/) was applied such that sequences below a minimum Phred quality value (QV) 17 scanning from 5' to the 3' end of the read are trimmed. This quality control is suggested in a Life technologies' white paper, "Methods, tools, pipelines for analysis of Ion PGM Sequencer miRNA and gene expression data" (https://tools.thermofisher.com/content/sfs/manuals/CO25176_0512.pdf). As a result, about 1% of bases were trimmed, and no read contains any ambiguous base 'N'. The data before and after quality control is summarized in Table S2. Finally, we sampled reads whose length is greater than 150 for training alignment parameters.

S2.3 Oxford Nanopore

In this study, we used 36 datasets reported in three publications (Quick *et al.*, 2014; Ashton *et al.*, 2015; Jain *et al.*, 2015), summarized in Table S5, where the version of Nanopore sequences are: E-coli K12 MG1655 (Quick *et al.*, 2014) is R7.3, *S.enterica* Typhi H58 (Ashton *et al.*, 2015) is R7, and M13mp18 phage (Jain *et al.*, 2015) is R7.3.

Table S2. Summary of IonTorrent read data utilized in this study

(i) Reference	(ii) Genome size	(iii) %GC	(iv) len.	(v) ave.len.	(vi) Depth	(vii) #reads
(before trimming)						
E-coli K12	4,639,675	50.79%	5–400	165.70	318	8,906,799
P.falciparum 3D7	23,270,305	19.36%	5–404	176.51	124	16,412,971
R.sphaeroides 241	4,602,977	68.79%	6–402	166.85	338	9,336,211
(after trimming)						
E-coli K12	4,639,675	50.79%	1–391	164.30	315	8,889,875
P.falciparum 3D7	23,270,305	19.36%	1–398	172.83	122	16,402,733
R.sphaeroides 241	4,602,977	68.79%	1–396	165.14	335	9,331,747

The meaning of each column is identical to Table S1 except for column (iv) that indicate the minimum and maximum read length.

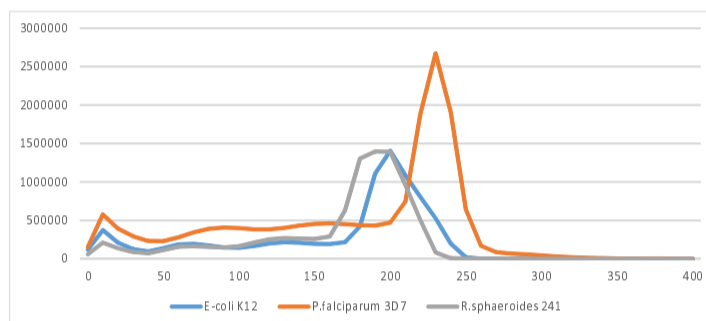


Fig. S2. Length distribution for reads generated by IonTorrent sequencers. See Table S2 for the details of the dataset.

Table S3. Mismatch, insertion and deletion rates reported in Bragg et al. (2013) Averaged values are shown among *Sulfolobus tokodaii*, *Bacillus amyloliquefaciens*, *Deinococcus maricopensis*.

Library	Mismatch	insertion	del
Ion Xpress 200	0.17%	2.69%	1.98%
Ion OneTouch 100	0.04%	0.84%	0.80%
Ion OneTouch 200	0.07%	1.76%	1.07%

Table S4. Mismatch and indel rates reported in Junemann et al. (2013)

Library	Mismatch	indel
Bioruptor 100	0.09%	0.35%
Ion Xpress 200	0.03%	0.40%
Ion Xpress 300	0.09%	0.71%
Ion Xpress 400	0.08%	0.67%

Oxford Nanopore sequencers produce several characteristic read types. The data is classified into 1D reads, 2D reads and HQ reads (high-quality). 1D reads are generated by one-pass, and are less accurate than 2D reads generated by 2-pass (Laszlo and Derrington, 2014). Also the dataset in Quick et al. (2014) includes high quality (HQ) reads, where the quality score of reads is higher than a threshold.

S3 An investigation of effects of data size for training

Although our training algorithm is accelerated by using the X-drop algorithm, it would require large computational cost if we handled all reads generated by sequencers. First of all, we investigate effects of data size used in training alignment parameters. In these experiments, we used various dataset sizes, where the dataset size means “the total length of trained sequences”. We tried sizes of 0.5 million (0.5M), 1M, 2M, 5M, 20M, 40M, and 80M in training alignment parameters for an E.coli K12 dataset generated by PacBio RS (P4-C2) (cf. Table S1). Each dataset was sampled randomly from the entire read set. For each data size we generated 5 datasets and trained alignment parameters with each.

The average and standard deviation of the trained parameters are shown in Table S6 (deletion/insertion gap costs) and Table S7 (substitution scores), indicating that a total length of 1M–10M is enough for training the parameters. These results also show that the training results are quite stable.

Moreover, we found that the trained parameters for local (Tables S6 and S7) and overlap alignments (Tables S8 and S9) are quite similar (as expected).

Table S5. Summary of Oxford Nanopore dataset

Quick 2014	E.coli K12 MG1655		1D	GIGADB		39,819	Ecoli_R73-f				
			1Dc			18,889	Ecoli_R73-r				
			2D			11,823	Ecoli_R73-2				
			1D (hq)			13,690	Ecoli_R73-hq-f				
			1Dc (hq)			13,690	Ecoli_R73-hq-r				
			2D (hq)			9,563	Ecoli_R73-hq-2				
Ashton 2014	S.enterica Typhi H58	H566_30_min_inc	1D	EBI	ERR668747	3,088	H566_30_min_inc-f				
			1Dc			1,678	H566_30_min_inc-r				
			2D			3,738	H566_30_min_inc-2				
			1D (hq)			236	H566_30_min_inc-hq-f				
			1Dc (hq)			236	H566_30_min_inc-hq-r				
			2D (hq)			92	H566_30_min_inc-hq-2				
		H566_ON_inc	1D	ERR668746	3,088	H566_ON_inc-f					
			1Dc		1,678	H566_ON_inc-r					
			2D		1,412	H566_ON_inc-2					
			1D (hq)		185	H566_ON_inc-hq-f					
			1Dc (hq)		185	H566_ON_inc-hq-r					
			2D (hq)		124	H566_ON_inc-hq-2					
			Jain 2015		M13mp18 phage	replicate1	2D (fail)	SRA	ERR732541	10,238	
							1Dc (fail)			ERR732542	30,937
1D (fail)	ERR732543	108,427									
2D (pass)	ERR732544	13,785									
1Dc (pass)	ERR732545	13,785									
1D (pass)	ERR732546	13,785									
replicate2	2D (fail)	ERR732547		14,308							
	1Dc (fail)	ERR732548		33,776							
	1D (fail)	ERR732549		92,968							
	2D (pass)	ERR732550		10,384							
	1Dc (pass)	ERR732551		10,384							
	1D (pass)	ERR732552		10,384							
replicate3	2D (fail)	ERR732553		4,822							
	1Dc (fail)	ERR732554		14,098							
	1D (fail)	ERR732555		52,135							
	2D (pass)	ERR732556		5,136							
	1Dc (pass)	ERR732557		5,136							
	1D (pass)	ERR732558		5,136							

The 1st column refers to the reference: Quick 2014 (Quick *et al.*, 2014), Ashton 2014 (Ashton *et al.*, 2015), and Jain 2015 (Jain *et al.*, 2015), whose version of Nanopore sequencers are R7.3, R7, and R7.3, respectively. 1D and 2D indicate one pass and two pass reads. HQ (high quality). 1Dc (1D read of a complementary sequence). fail (a read whose score is below threshold).

Table S6. Effects of training data size for estimating gap scores (with local alignment)

	deletion		insertion	
	existence	extension	existence	extension
0.5M	5.2 (0.39)	10.6 (0.48)	7 (0)	7 (0)
1M	5 (0)	10.4 (0.48)	7.2 (0.39)	6.6 (0.48)
2M	5 (0)	10.4 (0.48)	7 (0)	7 (0)
5M	5 (0)	10.8 (0.39)	7 (0)	7 (0)
10M	5 (0)	11 (0)	7 (0)	7 (0)
20M	5 (0)	11 (0)	7 (0)	7 (0)
40M	5 (0)	11 (0)	7 (0)	7 (0)
80M	5 (0)	11 (0)	7 (0)	7 (0)

The 1st column indicates the total length of sequences to be utilized for training alignment parameters. In this experiment, we used the E.coli K12 (P4-C2) dataset. Average (standard deviation) of each trained parameter is shown, where five random replicates were performed. See Table S8 for the results of overlap alignments.

S4 Training alignment parameters for PacBio, IonTorrent and Nanopore sequencers

In this subsection, we show results mainly for an E. coli genome (trained score matrix in Figure S3; trained gap costs in Table S10; results of alignment with trained parameters in Table S11), which was sequenced by PacBio, IonTorrent and Nanopore (Tables S1, S2, S5). We used 10M bp reads for training PacBio RS and IonTorrent (cf. previous section), while we used the entire read data for training Nanopore. Note that results for other genomes are shown in Section S9.

Table S7. Effects of data size for trained substitution score (with local alignment)

	A→A	A→C	A→G	A→T	C→A	C→C	C→G	C→T
0.5M	6 (0)	-19.6 (0.79)	-27 (1.41)	-29.2 (1.72)	-16 (0)	6 (0)	-22.6 (1.01)	-26 (1.67)
1M	6 (0)	-19 (0.63)	-25 (0.63)	-30 (1.09)	-16.2 (0.4)	6 (0)	-21.4 (0.48)	-25.2 (0.74)
2M	6 (0)	-19.2 (0.4)	-26.4 (0.8)	-30.8 (0.39)	-16 (0)	6 (0)	-21.8 (0.39)	-25.4 (0.48)
5M	6 (0)	-19.2 (0.4)	-26 (0)	-29.4 (0.48)	-16 (0)	6 (0)	-22 (0)	-25.4 (0.48)
10M	6 (0)	-19 (0)	-25.8 (0.39)	-29.2 (0.39)	-16 (0)	6 (0)	-22 (0)	-25.2 (0.4)
20M	6 (0)	-19 (0)	-26 (0)	-29.6 (0.48)	-16 (0)	6 (0)	-22 (0)	-25 (0)
40M	6 (0)	-19 (0)	-26 (0)	-29.2 (0.39)	-16 (0)	6 (0)	-22 (0)	-25.2 (0.4)
80M	6 (0)	-19 (0)	-26 (0)	-29 (0)	-16 (0)	6 (0)	-22 (0)	-25 (0)
	G→A	G→C	G→G	G→T	T→A	T→C	T→G	T→T
0.5M	-21.4 (0.8)	-17.8 (0.39)	6 (0)	-20.2 (0.4)	-23.6 (0.79)	-20.6 (0.79)	-22.4 (1.2)	6 (0)
1M	-22.2 (0.4)	-18.2 (0.4)	6 (0)	-20.8 (0.39)	-24.2 (0.74)	-20.6 (0.48)	-22.8 (0.74)	6 (0)
2M	-21.8 (0.74)	-18 (0)	6 (0)	-20.6 (0.48)	-23.8 (0.74)	-20.8 (0.39)	-22.6 (0.79)	6 (0)
5M	-21.8 (0.39)	-18 (0)	6 (0)	-20.8 (0.39)	-23.8 (0.39)	-20.6 (0.48)	-22.4 (0.48)	6 (0)
10M	-22 (0)	-18 (0)	6 (0)	-21 (0)	-24.2 (0.39)	-20.8 (0.39)	-23 (0)	6 (0)
20M	-22 (0)	-18 (0)	6 (0)	-21 (0)	-24 (0)	-20.8 (0.39)	-22.8 (0.39)	6 (0)
40M	-22 (0)	-18 (0)	6 (0)	-21 (0)	-24 (0)	-21 (0)	-23 (0)	6 (0)
80M	-22 (0)	-18 (0)	6 (0)	-21 (0)	-24 (0)	-21 (0)	-23 (0)	6 (0)

The 1st column indicates the total length of sequences to be utilized for training substitution scores. In this experiment, we used E.coli K12 (P4-C2) dataset. Average (standard deviation) of each trained parameter is shown, where five random replicates were performed. See Table S9 for overlap alignment.

Table S8. Effects of training data size for estimating gap scores (with overlap alignment)

	deletion		insertion	
	existence	extension	existence	extension
0.5M	4.6 (0.48)	10.8 (0.39)	6.4 (0.48)	8 (0.63)
1M	4.4 (0.48)	11 (0)	6.4 (0.48)	7.8 (0.4)
2M	4 (0)	11 (0)	6 (0)	8 (0)
5M	4.2 (0.4)	11 (0)	6.2 (0.39)	8 (0)
10M	4 (0)	11 (0)	6 (0)	8 (0)
20M	4 (0)	11 (0)	6 (0)	8 (0)
40M	4 (0)	11 (0)	6 (0)	8 (0)
80M	4 (0)	11 (0)	6 (0)	8 (0)

The 1st column indicates the total length of trained sequences. Average (standard deviation) of each trained parameter is shown, where five random replicates were performed. See Table S6 for local alignments.

S4.1 Comparisons among sequencers

By comparing results for a common genome (E.coli K12 MG1655), we performed a comparison among three sequencers in this subsection. For PacBio sequencers, datasets “P4-C2” (3rd row in Table S1) and “C2-C2” (1st row in Table S1) were taken. For Nanopore (Quick *et al.*, 2014), we used high-quality (HQ) 2D reads (dataset “Ecoli_R73-hq-2” in Table S5), which are expected to be more accurate than others (i.e., Ecoli_R73-f, Ecoli_R73-r and Ecoli_R73-2). For IonTorrent, dataset “E.coli K12” (after trimming) in Table S2 was used.

Figure S3 shows trained substitution score matrices for PacBio, IonTorrent and Nanopore. The results indicate that the mismatch costs follow this order: (larger) IonTorrent > PacBio > Nanopore (smaller), suggesting that IonTorrent sequencers have fewer substitution errors than PacBio and Nanopore sequencers. On the other hand, Table S10 shows trained affine gap costs; Interestingly, gap existence costs are larger than gap extension costs for Nanopore, while gap existence costs are smaller than gap extension costs for PacBio and IonTorrent, suggesting that Nanopore sequencers tend to include more multi-base gaps than the others.

Table S11 shows the statistics of alignment results with the trained parameters, where the rates of mismatch, insertion and deletion are computed based on them. The results reproduce known characteristics of error profiles of each sequencer (e.g., PacBio reads include more insertions than substitutions and deletions). The results also show that aligned rate for *overlap* alignment of PacBio and Nanopore is less than 42%, indicating that many reads could not be aligned end-to-end to the reference genome (with the default X-drop setting). For this dataset, the accuracy of Nanopore 2D reads is much better than Nanopore 1D reads. (Specifically, the mismatch rate for 1D reads is quite bad (21.1%)). There is a small difference between 2D-HQ and 2D reads; the aligned rates are slightly improved for high quality reads.

S4.2 PacBio RS II

We trained alignment parameters for 11 datasets (Table S1). Training results for each dataset are shown in Figures S5 (substitution score matrix with local alignments), Figure S6 (substitution score matrix with overlap alignments) and Table S15 (gap existence and extension costs). In addition, we mapped

Table S9. Effects of data size for trained substitution scores (with overlap alignment)

	A→A	A→C	A→G	A→T	C→A	C→C	C→G	C→T
0.5M	6 (0)	-21.6 (1.35)	-28.2 (1.72)	-30.8 (1.16)	-16 (0)	6 (0)	-23.8 (2.13)	-26.8 (2.78)
1M	6 (0)	-20.6 (1.01)	-27.8 (1.59)	-32.2 (2.03)	-16.8 (0.39)	6 (0)	-22.8 (1.16)	-26.4 (1.35)
2M	6 (0)	-20.8 (0.39)	-28.8 (0.74)	-34.4 (1.2)	-16.8 (0.39)	6 (0)	-22.4 (0.48)	-27 (1.09)
5M	6 (0)	-20.6 (0.48)	-29.8 (0.74)	-34.2 (2.03)	-16.2 (0.4)	6 (0)	-22.6 (0.48)	-27.2 (0.4)
10M	6 (0)	-20.4 (0.48)	-29.2 (0.74)	-34 (0.89)	-16 (0)	6 (0)	-23 (0)	-27.4 (0.48)
20M	6 (0)	-20.8 (0.39)	-29 (0)	-35.4 (0.8)	-16.2 (0.4)	6 (0)	-23 (0)	-27 (0)
40M	6 (0)	-20.6 (0.48)	-29 (0)	-34.4 (1.49)	-16 (0)	6 (0)	-22.8 (0.39)	-27.2 (0.4)
80M	6 (0)	-21 (0)	-29 (0)	-34.4 (0.48)	-16 (0)	6 (0)	-23 (0)	-27 (0)

	G→A	G→C	G→G	G→T	T→A	T→C	T→G	T→T
0.5M	-23.4 (0.8)	-19.4 (0.48)	6 (0)	-21 (1.09)	-24.8 (1.32)	-22.2 (1.16)	-23.4 (0.8)	6 (0)
1M	-24.4 (1.2)	-19.2 (0.4)	6 (0)	-21.8 (1.16)	-27.2 (1.46)	-22.2 (0.74)	-25.8 (2.13)	6 (0)
2M	-23.8 (0.74)	-19.2 (0.4)	6 (0)	-21 (0)	-26 (0.63)	-22.6 (0.48)	-24.6 (1.35)	6 (0)
5M	-24 (0)	-19.2 (0.4)	6 (0)	-21.4 (0.8)	-26.6 (0.48)	-22.2 (0.4)	-25 (0.63)	6 (0)
10M	-24 (0.63)	-19 (0)	6 (0)	-21.2 (0.4)	-26.2 (0.74)	-22.2 (0.4)	-25 (0.63)	6 (0)
20M	-24 (0)	-19 (0)	6 (0)	-21.4 (0.48)	-26.4 (0.48)	-22.2 (0.4)	-24.4 (0.48)	6 (0)
40M	-24 (0)	-19 (0)	6 (0)	-21.2 (0.4)	-26.2 (0.4)	-22 (0)	-24.6 (0.48)	6 (0)
80M	-24 (0)	-19 (0)	6 (0)	-21 (0)	-26 (0)	-22 (0)	-25 (0)	6 (0)

The 1st column indicates the total length of sequences for using training. Average (standard deviation) of each trained parameter is shown, where five random replicates were performed. See Table S7 for local alignments.

	(a) PacBio (P4-C2)				(b) IonTorrent				(c) Nanopore (2D, HQ)						
	A	C	G	T	A	C	G	T	A	C	G	T			
Overlap alignment	A	6	-21	-29	-34	A	6	-35	-31	-35	A	6	-14	-11	-24
	C	-16	6	-23	-28	C	-35	6	-37	-32	C	-11	6	-10	-11
	G	-24	-19	6	-22	G	-31	-36	6	-36	G	-9	-9	6	-11
	T	-26	-22	-24	6	T	-35	-31	-37	6	T	-23	-12	-13	6
Local alignment	A	6	-19	-26	-29	A	6	-37	-32	-37	A	6	-12	-10	-22
	C	-16	6	-22	-25	C	-37	6	-39	-33	C	-10	6	-9	-10
	G	-22	-18	6	-21	G	-32	-39	6	-38	G	-8	-8	6	-10
	T	-24	-21	-23	6	T	-37	-32	-39	6	T	-21	-11	-12	6

Fig. S3. Trained substitution score matrices for PacBio, IonTorrent, and Nanopore sequencers, with *E. coli* K12 DNA. In each matrix, query letters correspond to columns and reference letters correspond to rows.

Table S10. Comparison of trained affine gap costs for PacBio, IonTorrent and Nanopore sequencers

	Reference	(a) local alignment				(b) overlap alignment			
		deletion cost		insertion cost		deletion cost		insertion cost	
		existence	extension	existence	extension	existence	extension	existence	extension
PacBio (P4-C2)	<i>E. coli</i> K12	5	11	7	7	4	11	6	8
IonTorrent	<i>E. coli</i> K12	8	16	10	15	9	15	12	12
Nanopore (2D, HQ)	<i>E. coli</i> K12	11	4	14	3	11	4	14	3

The genome utilized in this result is *E. coli* K12. Trained gap costs with (a) local and (b) overlap alignments are shown. The gap cost for the 1st gap is equal to “existence + extension” cost, while the one for later gaps is equal to “extension” cost in our model. For Nanopore, we utilized the “Ecoli_R73-hq-2” dataset in Table S5.

reads to the reference genome using LAST with the *trained* parameters, and, for comparison, we also mapped the reads using BLASR and LAST with *manual* parameters. Statistics of alignment (mapping) with the trained alignment parameters are shown in Table S16, and various analyses for mapped reads are shown in Figure S4 (for *E. coli* K12) and Figures S10–S15 (for all datasets).

In summary, we obtained the following observations, most of which are consistent with (or suggested in) previous studies. Similar trained parameters have been obtained between using either local or overlap alignments (Table S15 and Figures S5, S6). Substitution error rates are smaller than indel error rates, and insertion error rates are larger than deletion error rates (Table S11 and Table S16). For example, mismatch, insertion and deletion rates are 1.1%, 10.1%, and 2.9%, respectively, for PacBio (C2-C2) with local alignments. The substitution error rates are not improved for newer technology, while the indel error rates are greatly improved (Table S16). For the latest version of PacBio, P6-C4, the difference between insertion and deletion rates is smaller than for the former versions (in most cases). For example, insertion rate is 4.7% and deletion rate is 3.9% for P6-C4 (Table S16). The sizes of

Table S11. Results of alignment (mapping) with trained alignment parameters for PacBio RS, IonTorrent and Nanopore sequencers

	Reference	(a) local alignment					(b) overlap alignment				
		aligned rate		substitution rate			aligned rate		substitution rate		
		seq	base	mismatch	ins	del	seq	base	mismatch	ins	del
PacBio (C2–C2)	E-coli K12	96.5	75.1	1.1	10.1	2.9	53.2	42.0	1.0	9.9	2.8
PacBio (P4–C2)	E-coli K12	99.7	73.0	0.8	6.6	3.6	39.8	29.6	0.7	6.1	3.5
IonTorrent	E-coli K12	98.5	97.8	0.0	0.4	0.5	98.0	98.0	0.1	0.5	0.6
Nanopore (2D, HQ)	E-coli K12	86.2	66.9	5.8	6.2	9.8	34.9	30.0	5.0	6.8	8.6
Nanopore (2D)	E-coli K12	75.5	65.5	5.7	6.6	10.7	29.0	26.3	5.0	6.8	8.7
Nanopore (1D)	E-coli K12	51.0	52.3	21.1	3.3	14.6	29.1	28.6	19.6	3.8	13.9

In these experiments, we tried (a) local alignments and (b) overlap alignments for training. The 3rd and 4th (8th and 9th) columns show aligned rates (%) with respect to sequence and base (nucleotide), respectively. The 5th, 6th and 7th (10th, 11th and 12th) columns indicate mismatch, insertion and deletion rates (%), respectively. Nanopore (2D, HQ), Nanopore (2D) and Nanopore (1D) show datasets Ecoli_R73-hq-2, Ecoli_R73-2, and Ecoli_R73-f, respectively, in Table S5.

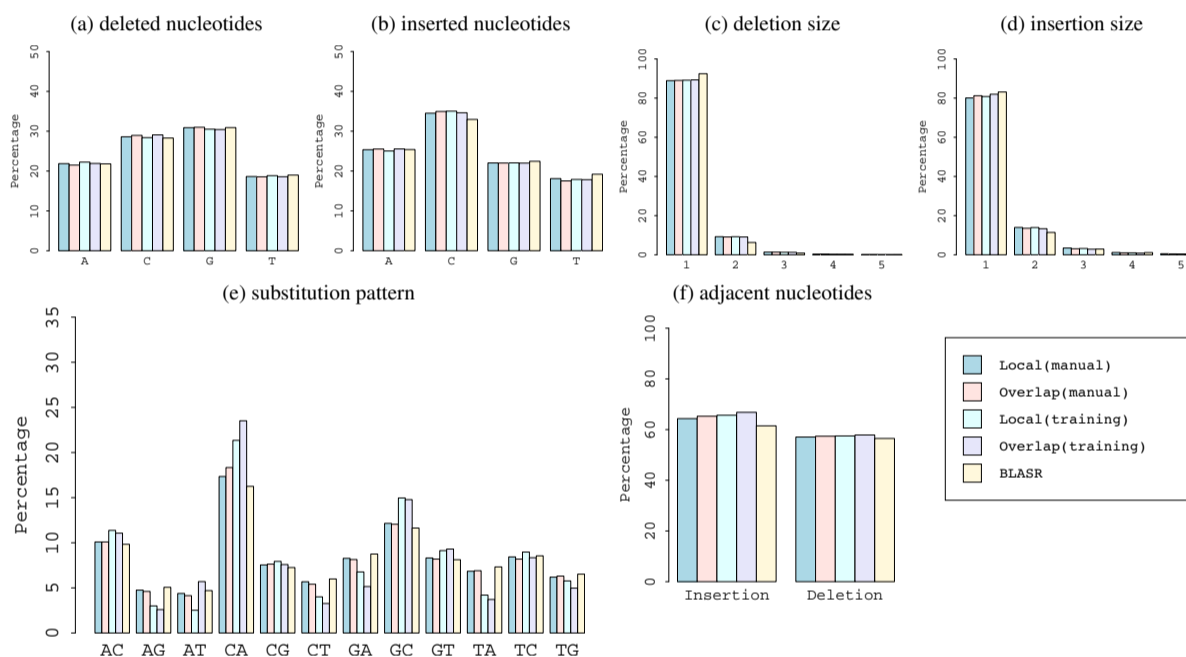


Fig. S4. Summary of alignment results for PacBio (P4–C2) on E.coli K12 for LAST with trained/manual alignment parameters and BLASR (Chaisson and Tesler, 2012). (a, b) the frequency of deleted and inserted nucleotides; (c, d) the size of deletion or insertions; (e) the frequency of base substitutions; (f) the frequency of adjacent nucleotides that are identical with the deleted/inserted nucleotide. See Figures S10, S11, S12, S13, S14, and S15 for the other datasets. For a set of bars in each figure, LAST (local alignments) with traditional scores (Ono et al., 2013) (blue), LAST (overlap alignments) with traditional scores (Ono et al., 2013) (light red), LAST (local alignments) with trained scores (light blue) LAST (overlap alignments) with trained scores (purple), and BLASR (yellow) are shown from left to right.

indels seem to follow a Poisson distribution, where the size of most (more than 80%) of indels is 1 (Figures S11 and S13). This is reflected in trained gap costs, where gap extension costs are larger than gap existence costs (Table S15). The proportion of inserted (deleted) bases that are identical to adjacent bases (50–70%) is larger than the random expectation (44%) (Figure S15). For the distribution of sizes of indels, BLASR includes slightly more indels of size 1 and fewer indels of size 2 than LAST with trained parameters (Figures S11 and S13). In Figure S14, the percentages of A to T substitution for overlap alignments with respect to V.cholerae N5 and C.elegance are higher than the other substitutions; we did not find any clear reason, although we have investigated the results.

S4.3 IonTorrent

IonTorrent (including Ion PGM and Ion Proton sequencers) is one of the 4th-generation sequencers, whose Ion Xpress 200 was used in (Ross et al., 2013); these datasets are described in Table S2. We trained alignment parameters for these 3 datasets, and show these parameters (score matrix and gap costs) in Figure S3b, Figure S7, Table S10b, and Table S17.

Similar to the PacBio results in the previous section, there is no difference between trained parameters using local and overlap alignments. IonTorrent sequencers, in general, achieved lower error rates than PacBio and Nanopore, while the reads are shorter (the length distribution of IonTorrent reads is shown in Figure S2). Additionally, the substitution error rates are lower than the indel error rates (Table S18), consistent with previous studies (Tables S3 and S4 (Bragg et al., 2013; Junemann et al., 2013)).

S4.4 Nanopore

We trained alignment parameters for 36 Nanopore datasets described in Table S5. Figures S8 and S9 show substitution score matrices for each dataset using local and overlap alignment, respectively, and Table S19 shows the trained gap costs. Also we show the results of alignment with trained parameters in Table S20.

These results show that similar trained parameters were obtained for local and overlap alignments. As expected, mismatch costs and gap costs for 2D reads are stronger than those of 1D reads. Table S20 indicates alignment results for Nanopore datasets using the trained parameters, showing that quality of the 2D reads is much better than 1D reads. However, the total error rate of 2D (high-quality) reads is approximately 18%, indicating that there is room to improve the accuracy. Moreover, trained gap (existence and extension) costs are shown in Table S19. The results show that the insertion cost (existence + extension) is larger than the deletion cost, suggesting that Nanopore reads tend to include more deletions than insertions. (Note that PacBio reads contain more insertions than deletions.)

S5 Evaluation of alignment accuracy with simulated dataset

S5.1 Simulated data

Simulated data were generated by PBSIM with model based simulation (Ono *et al.*, 2013). The parameters of PBSIM were taken from statistics of H. sapiens PacBio read data (P5-C3), where the ratio of sequencing errors is substitution : insertion : deletion = 1 : 9 : 5. Specifically, the command line of PBSIM is as follows.

```
pbsim --data-type CLR --depth 0.01 --model_qc model_qc_clr
--length-mean 7561 --length-sd 5895 --accuracy-mean 0.85 --accuracy-sd 0.02
--length-min 502 --length-max 35488 --difference-ratio 1:9:5 hg19.fasta
```

Finally, 1000 sequences are randomly selected from the simulated reads, which are converted to FASTA format, for our evaluation. The summary of our data is shown as follows:

	real data	simulated data	randomly selected data for evaluation
read_num	22081	4266	1000
read_len_mean	7560.6	7256.8	7275.6
read_len_sd	5895.2	5066.0	5003.3
read_len_min	502	502	502
read_len_max	35488	34920	34167
read_accuracy_mean	84.49%	84.94%	85.01%
read_accuracy_sd	1.91%	2.00%	1.97%

S5.2 Training parameters with LAST-TRAIN

We trained alignment parameters using the 1000 sequences from Section S5.1, by the following command line: `last-train -T0 hg19 reads.fasta` for local alignment, and `last-train -T1 hg19 reads.fasta` for overlap alignment.

As a result, the trained score matrices and gap costs are

	A	C	G	T		A	C	G	T
A	5	-16	-16	-18	A	5	-16	-16	-18
C	-18	7	-17	-18	C	-18	7	-17	-18
G	-18	-17	7	-18	G	-18	-17	7	-18
T	-19	-16	-16	5	T	-19	-16	-16	5

and

	deletion costs		insertion costs	
	existence	extention	existence	extention
local	0	15	2	11
overlap	0	15	2	11

respectively. The computational times for training are 19,296 and 2,849 seconds for local and overlap alignments, respectively. Note that exactly identical alignment parameters were obtained for local and overlap alignments. Also note that the deletion existence cost indicates linear gap costs in this case.

S5.3 Alignment of reads to reference genome with lastal

Alignments of reads to the reference genome were performed with the following command lines:

1. manual parameters (match score $r=1$, mismatch cost $q=1$, gap exist cost $a=1$, gap extend cost $b=1$):

```
lastal -T0 (or 1) -r1 -q1 -a1 -b1 -m100 hg19 read.fasta | last-split -m1
```

2. manual parameters (match score $r=1$, mismatch cost $q=2$, gap exist cost $a=1$, gap extend cost $b=1$):

```
lastal -T0 (or 1) -r1 -q2 -a1 -b1 -m100 hg19 read.fasta | last-split -m1
```

3. trained parameters

```
lastal -T0 (or 1) -plast-train[trained result] -m100 hg19 read.fasta | last-split -m1
```

In the above, alignments are equivalent to Viterbi alignments (i.e. maximum score alignments). In addition, we tried probabilistic alignments, where `-j[5 or 6]` `-g[0.5, 1, 2 or 4]` are added to the `lastal` options in the above command lines, where `-j5` and `-j6` specify γ -centroid and LAMA alignments, respectively, and `-g` specifies the parameter for adjusting sensitivity and PPV. See Hamada *et al.* (2011) for the details.

S5.4 Results

The alignment accuracy of conventional (Viterbi) alignments are shown in Table S12, where the evaluation was performed based on sensitivity and PPV for aligned columns with respect to estimated and correct alignments (note that correct alignments are available because the reads are simulated).

This table indicates that trained parameters present better performance than two types of manual parameters. Interestingly, computational time with trained parameters are faster than those of manual parameters. This is because the manual parameters have weak mismatch and gap costs, causing `lastal` to explore alignment extensions further.

The results of probabilistic alignments are also shown in Table S13, indicating that probabilistic alignments with trained parameters achieve the best performance.

Table S12. Alignment accuracy for Viterbi alignments with manual and trained parameters

		sensitivity	PPV	time (s)
manual ($q = 1$)	local	83.7299%	85.5629%	13601
	overlap	83.7351%	85.5614%	8339
manual ($q = 2$)	local	85.2853%	86.5776%	3076
	overlap	85.2911%	86.5764%	2026
trained	local	85.3665%	86.6776%	1295
	overlap	85.3709%	86.6773%	979

S6 Haplotype phasing test

See Section 3 in the main manuscript for the details. Table S14 is an additional table referred in the main manuscript.

S7 Command lines

S7.1 Mapping PacBio reads

On PacBio RS, we compared LAST with trained parameters to the following:

- LAST local alignments and overlap alignments with the *manual* parameters previously reported in Ono *et al.* (2013), where the parameters were manually optimized for PacBio reads.
- BLASR (Chaisson and Tesler, 2012), which is a specialized mapper for PacBio reads, with options: `-minPctIdentity=70.0`, `-maxScore=-1000`.

For LAST, the alignments were filtered using `last-map-probs` with default parameters, to get at most one confident unique-best alignment per query. For BLASR, the alignment with the best score, and non-overlapping alignments were greedily chosen for evaluation.

S7.2 Haplotype phasing

In the results of haplotype phasing (described in the Results section in the main manuscript) we utilized the following command lines.

1. LAST (manual parameters, $q = 1$)
`lastal -r1 -q1 -a1 -b1 -m100 genome long-read`
2. LAST (manual parameters, $q = 2$)
`lastal -r1 -q2 -a1 -b1 -m100 genome long-read`
3. LAST (training parameters using LAST-TRAIN)
`lastal -plast-train-results -m100 genome long-read`
last-train-results are obtained by the following: `last-train genome long-read`
4. GraphMap
`graphmap align -r genome -d long-read`

Table S13. Alignment accuracy for probabilistic alignments with manual and trained parameters

		-j	-g	sensitivity	PPV	time (s)	# alignments	
manual ($q = 1$)	local	5	0.5	76.7157%	72.4103%	66422	2303	
		5	1	84.6192%	83.3576%	66595	1101	
		5	2	86.2199%	87.6147%	256562	1007	
		5	4	86.2099%	87.7949%	66755	1000	
		6	0.5	86.2847%	87.8355%	311626	1000	
		6	1	86.3041%	87.8351%	69628	1000	
		6	2	86.3099%	87.8342%	140735	1000	
		6	4	86.3107%	87.8337%	72961	1000	
		overlap	5	0.5	76.8428%	72.4944%	137305	2256
	5		1	84.6286%	83.3594%	13102	1097	
	5		2	86.2220%	87.6144%	10219	1007	
	5		4	86.2117%	87.7953%	11824	1000	
	6		0.5	86.3125%	87.8343%	207090	1000	
	6		1	86.3123%	87.8341%	14697	1000	
	6		2	86.3123%	87.8341%	14558	1000	
	6		4	86.3122%	87.8340%	234692	1000	
	manual ($q = 2$)		local	5	0.5	79.9906%	76.3362%	18935
		5		1	86.1485%	85.6906%	14095	1158
5		2		87.1314%	88.3498%	14445	1015	
5		4		87.1376%	88.4284%	18387	1000	
6		0.5		87.0432%	88.2847%	14783	1000	
6		1		87.0527%	88.2835%	14869	1000	
6		2		87.0548%	88.2830%	17139	1000	
6		4		87.0552%	88.2830%	14968	1000	
overlap		5		0.5	80.4955%	76.7712%	172452	2873
		5	1	86.1667%	85.7154%	195656	1129	
		5	2	87.1346%	88.3522%	4258	1011	
		5	4	87.1394%	88.4291%	190012	1000	
		6	0.5	87.0571%	88.2844%	198164	1000	
		6	1	87.0570%	88.2841%	206900	1000	
		6	2	87.0568%	88.2839%	251661	1000	
		6	4	87.0568%	88.2838%	4113	1000	
		trained	local	5	0.5	58.0370%	77.8066%	196937
5				1	86.3640%	86.5191%	7653	1834
5	2			87.3598%	88.6375%	187663	1026	
5	4			87.3655%	88.6875%	7636	1000	
6	0.5			87.3314%	88.6453%	200976	1000	
6	1			87.3420%	88.6446%	135645	1000	
6	2			87.3451%	88.6444%	10594	1000	
6	4			87.3452%	88.6444%	8349	1000	
overlap	5			0.5	61.4243%	82.6763%	202655	22134
	5		1	86.4218%	86.5787%	199264	1786	
	5		2	87.3625%	88.6403%	5119	1023	
	5		4	87.3669%	88.6886%	198928	1000	
	6		0.5	87.3468%	88.6458%	286041	1000	
	6		1	87.3469%	88.6458%	6990	1000	
	6		2	87.3469%	88.6458%	5411	1000	
	6		4	87.3469%	88.6458%	5547	1000	

Column -j shows the type of probabilistic alignment, where 5 indicates γ -centroid alignment and 6 indicates LAMA alignment (Hamada *et al.*, 2011).

S8 Discussion

S8.1 Comparison of LAST-TRAIN and MarginAlign

MarginAlign (Jain *et al.*, 2015) is a method to obtain accurate sequence alignments: it has several similarities to LAST-TRAIN, so here we discuss a comparison of these two methods. Both train alignment parameters using heuristically-accelerated expectation-maximization. As described above, LAST performs a forward-backward algorithm within an X-drop region of the dynamic programming matrix (Frith *et al.*, 2010b), whereas MarginAlign performs

Table S14. Known haplotype for CYP2D6

	rs35742686	rs3892097
CYP2D6*3	–	C
CYP2D6*4	T	T
(Reference: hg19)	T	C

The genome of our sample (NA12878) is considered to be the diploid of CYP2D6*3/*4 (Numanagi *et al.*, 2015; Twist *et al.*, 2016).

this algorithm within a band around a guide alignment (found with an aligner such as LAST). LAST-TRAIN re-aligns the sequences from scratch at each iteration, whereas MarginAlign uses fixed guide alignments. LAST-TRAIN outputs substitution and gap scores, which can in principle be used by any aligner, whereas MarginAlign trains its internal model and outputs alignments.

The main difference is that MarginAlign “polishes” crude initial alignments, whereas LAST-TRAIN finds parameters that enable accurate from-scratch alignments. Thus, MarginAlign relies on the initial alignments being approximately correct (e.g. not paralogous), whereas LAST-TRAIN has no such limitation.

This difference is especially important when we use LAST-SPLIT, which aligns each part of a query sequence to its best match in the reference, allowing for the possibility that different parts of one query may match disjoint loci (which is common for long queries). For this to work well (i.e. accurately align orthologous bases), the alignment parameters should be trained before rather than after alignment.

Another similarity is that LAST has options to find “marginal” alignments. This involves calculating the probability that each alignment column is correct, by marginalizing over the probabilities of all alignments that include this column. To the best of our knowledge, the first published variant of marginal alignment was “centroid alignment” (Miyazawa, 1995). Another variant is “alignment metric accuracy” (Schwartz *et al.*, 2005), which is used by MarginAlign. LAST can use either “gamma-centroid alignment” (a generalization of centroid alignment) (Frith *et al.*, 2010b), or “LAMA alignment” (a generalization of AMA to local alignment) (Hamada *et al.*, 2011).

A final difference is that we believe LAST/LAST-TRAIN is easier to install, because it has weaker dependencies. We could not easily install MarginAlign on any of our computers.

S8.2 Resisting the temptation of over-alignment

It is tempting to judge alignment success by the fraction of DNA reads, and bases therein, that get aligned. In particular, it has been suggested that alignments covering a larger portion of each DNA read are better. Do alignments with our trained parameters tend to maximize coverage of each read? The answer is clearly “no”, because we can *trivially* increase coverage by weakening the mismatch or gap costs (Frith *et al.*, 2008). This creates a risk of a race-to-the-bottom in alignment stringency, which should be resisted.

There are further reasons to not extend alignments too aggressively. It has been suggested that PacBio reads include long stretches (hundreds or thousands of bases) that are essentially junk (<https://dazzlerblog.wordpress.com/2015/11/06/intrinsic-quality-values/>). Also, longer reads are more likely to overlap rearrangements, such as inversions. This means that the correct alignment of a read may consist of multiple discontinuous pieces: LAST-SPLIT is designed to align such reads (Frith and Kawaguchi, 2015). We should be wary of aggressive methods such as *forcing* alignment between co-linear anchors (Frith and Kawaguchi, 2015): instead, we can let the statistical model tell us when the sequences cease to be similar.

S8.3 LAST-TRAIN and sequence quality data

LAST has an option (not used in this study) to incorporate sequence quality data, which is often available in fastq format (Frith *et al.*, 2010a; Hamada *et al.*, 2011). Such quality data indicates the error probability of each base. LAST assumes that it indicates substitution (not indel) error probabilities, which may not always be the case. Anyhow, the score for aligning a query base to a reference base is derived from both the substitution score matrix and the quality data. The matrix reflects the frequencies of *real* substitutions (not errors), and thus specifies the scores for aligning high-quality bases: these scores are adjusted for low-quality bases.

Currently, last-train cannot train substitution parameters while considering sequence quality data. Often, real substitutions are expected to be rare, e.g. if we align human DNA reads to a human genome. In such cases, it is appropriate to use a stringent score matrix with mismatch cost 3–4 times the match score (States *et al.*, 1991). LAST-TRAIN currently has a useful (but not ideal) option to train insertion and deletion parameters using fastq data: in this case the score matrix is fixed at its initial (typically stringent) state.

LAST’s method for incorporating fastq quality data has a disadvantage. On one hand, it usefully considers that (say) a low-quality t is not unlikely to really be an a, c, or g. On the other hand, it cannot consider that (say) it is less likely to be an a, which would be helpful for nanopore because a:t substitutions are rare. Actually, LAST internally uses 4 probabilities (of being a, c, g, or t) per base, and has an option to read such data, which can eliminate the disadvantage. What is lacking is a convenient way to convert fastq to this format.

In summary, it is unfortunately not clear whether it is best to use sequence quality data, and it probably depends on the type of data.

S9 Trained parameters and statistics for each sequencer

(a) E.coli K12 (C2-C2)				(b) V.cholerae N5 (C2-C2)				(c) E.coli K12 (XL-C2)				(d) M.ruber (XL-C2)							
A	C	G	T	A	C	G	T	A	C	G	T	A	C	G	T				
A	6	-13	-23	-36	A	6	-12	-20	-28	A	6	-20	-26	-53	A	8	-23	-29	-30
C	-19	6	-20	-29	C	-17	6	-20	-26	C	-18	6	-20	-46	C	-15	5	-21	-23
G	-23	-16	6	-24	G	-20	-15	6	-19	G	-21	-18	6	-24	G	-25	-24	5	-21
T	-22	-15	-18	6	T	-19	-14	-17	6	T	-20	-17	-16	6	T	-26	-25	-23	8

(e) P.heparinus (XL-C2)				(f) E.coli K12 (P4-C2)				(g) S.cerevisiae (P4-C2)				(h) N.creassa (P4-C3)							
A	C	G	T	A	C	G	T	A	C	G	T	A	C	G	T				
A	6	-16	-26	-34	A	6	-19	-26	-29	A	5	-16	-22	-39	A	6	-16	-18	-28
C	-19	7	-22	-29	C	-16	6	-22	-25	C	-16	7	-22	-25	C	-15	6	-20	-21
G	-24	-19	7	-24	G	-22	-18	6	-21	G	-19	-16	8	-19	G	-17	-15	6	-20
T	-24	-20	-17	6	T	-24	-21	-23	6	T	-22	-15	-18	5	T	-18	-13	-17	6

(i) H.sapiens (P5-C3)				(j) D.melanogaster (P5-C3)				(k) C.elegans (P6-C4)						
A	C	G	T	A	C	G	T	A	C	G	T			
A	6	-16	-26	-34	A	5	-15	-17	-22	A	5	-13	-20	-23
C	-19	7	-22	-29	C	-15	7	-14	-17	C	-16	8	-19	-19
G	-24	-19	7	-24	G	-18	-21	7	-19	G	-21	-25	8	-22
T	-24	-20	-17	6	T	-22	-20	-21	6	T	-23	-20	-20	5

Fig. S5. Trained score matrix of substitution for PacBio RS sequencers with local alignment. See Table S1 for the utilized datasets in detailed. Figure S6 shows the corresponding score matrix with respect to overlap alignment.

(a) E.coli K12 (C2-C2)				(b) V.cholerae N5 (C2-C2)				(c) E.coli K12 (XL-C2)				(d) M.ruber (XL-C2)							
A	C	G	T	A	C	G	T	A	C	G	T	A	C	G	T				
A	6	-14	-25	-41	A	6	-17	-24	-35	A	6	-20	-27	-51	A	8	-24	-31	-32
C	-19	6	-21	-30	C	-18	6	-22	-27	C	-18	6	-21	-45	C	-16	5	-22	-24
G	-26	-17	6	-25	G	-22	-18	6	-22	G	-23	-19	6	-24	G	-26	-25	5	-21
T	-24	-16	-19	6	T	-22	-17	-19	6	T	-20	-17	-16	6	T	-28	-26	-24	8

(e) P.heparinus (XL-C2)				(f) E.coli K12 (P4-C2)				(g) S.cerevisiae (P4-C2)				(h) N.creassa (P4-C3)							
A	C	G	T	A	C	G	T	A	C	G	T	A	C	G	T				
A	6	-18	-27	-36	A	6	-21	-29	-34	A	5	-18	-22	-45	A	6	-18	-26	-41
C	-19	7	-22	-30	C	-16	6	-23	-28	C	-18	7	-21	-26	C	-16	6	-23	-27
G	-26	-20	7	-24	G	-24	-19	6	-22	G	-21	-18	8	-21	G	-20	-17	6	-21
T	-26	-20	-18	6	T	-26	-22	-24	6	T	-24	-17	-19	5	T	-21	-17	-19	6

(i) H.sapiens (P5-C3)				(j) D.melanogaster (P5-C3)				(k) C.elegans (P6-C4)						
A	C	G	T	A	C	G	T	A	C	G	T			
A	6	-18	-27	-36	A	6	-16	-25	-29	A	5	-14	-24	-28
C	-19	7	-22	-30	C	-16	7	-15	-22	C	-16	8	-19	-21
G	-26	-20	7	-24	G	-25	-25	7	-22	G	-25	-26	8	-23
T	-26	-20	-18	6	T	-31	-37	-25	6	T	-28	-24	-22	5

Fig. S6. Trained score matrix of substitution for PacBio RS sequencers (overlap alignment). See Table S1 for the utilized datasets in detailed.

Table S15. Trained affine gap costs for PacBio RS II sequencers

Reference	local alignment				overlap alignment				
	deletion cost		insertion cost		deletion cost		insertion cost		
	existence	extension	existence	extension	existence	extension	existence	extension	
C2-C2	E.coli K12	7	9	7	6	6	10	6	6
	V.cholerae N5	7	9	7	6	6	10	6	7
XL-C2	E.coli K12	4	11	6	8	4	11	5	8
	M.ruber	5	12	7	7	4	12	7	7
	P.heparinus	6	11	7	7	6	12	6	7
P4-C2	E.coli K12	5	11	7	7	4	11	6	8
	S.cerevisiae	6	10	6	7	5	11	6	8
P4-C3	N.creassa	6	9	7	6	5	10	6	7
P5-C3	D.melanogaster	6	10	6	7	4	12	6	8
	H.sapiens	5	8	9	6	6	8	9	6
P6-C4	C.elegans	5	10	7	8	5	10	6	9

The gap cost for the 1st gap is equal to existence + extension. local and overlap alignments

Table S16. Results of alignment (mapping) results with trained alignment parameters for PacBio RS

Reference	local alignment						overlap alignment				
	aligned rate		substitution rate				aligned rate		substitution rate		
	seq	base	mismatch	ins	del	seq	base	mismatch	ins	del	
C2-C2	E.coli K12	96.5	75.1	1.1	10.1	2.9	53.2	42.0	1.0	9.9	2.8
	V.cholerae N5	92.5	44.0	1.3	8.9	3.7	2.5	2.3	1.0	8.6	3.3
XL-C2	E.coli K12	99.6	81.9	1.0	7.7	3.7	58.1	47.9	1.0	7.7	3.6
	M.ruber	99.3	86.1	0.7	6.6	2.8	66.9	60.1	0.6	6.5	2.8
	P.heparinus	99.5	84.5	0.7	8.2	2.2	65.7	57.0	0.7	8.0	2.2
P4-C2	E.coli K12	99.7	73.0	0.8	6.6	3.6	39.8	29.6	0.7	6.1	3.5
	S.cerevisiae	94.6	71.9	1.0	7.3	3.7	40.0	29.6	0.9	7.3	3.3
P4-C3	N.creassa	99.1	62.5	1.3	9.3	3.8	31.5	19.5	1.1	8.4	3.8
P5-C3	D.melanogaster	92.1	66.2	1.0	6.9	3.1	43.6	35.2	0.8	6.9	3.1
	H.sapiens	97.9	79.7	1.8	5.4	6.1	57.2	45.3	2.2	5.3	5.7
P6-C4	C.elegans	96.6	77.7	1.0	4.7	3.9	47.1	37.2	0.9	4.7	3.9

The 3rd and 4th (8th and 9th) columns show aligned rates for sequence level and base level, respectively.

	(a) E.coli K12				(b) P.falciparum 3D7				(c) R.sphaeroides 241						
	A	C	G	T	A	C	G	T	A	C	G	T			
overlap alignment	A	6	-35	-31	-35	A	4	-29	-27	-29	A	8	-35	-32	-36
	C	-35	6	-37	-32	C	-28	10	-30	-28	C	-36	5	-36	-31
	G	-31	-36	6	-36	G	-25	-29	10	-27	G	-32	-35	5	-35
	T	-35	-31	-37	6	T	-28	-28	-31	4	T	-37	-32	-36	8
local alignment	A	6	-37	-32	-37	A	4	-30	-28	-29	A	8	-37	-33	-39
	C	-37	6	-39	-33	C	-29	10	-32	-29	C	-37	5	-38	-32
	G	-32	-39	6	-38	G	-26	-31	10	-28	G	-33	-37	5	-37
	T	-37	-32	-39	6	T	-29	-29	-32	4	T	-39	-33	-38	8

Fig. S7. Trained score matrices of substitution for IonTorrent sequencers with three reference genomes

Table S17. Trained affine gap cost for ION Torrent dataset

Reference	local alignment				overlap alignment			
	deletion cost		insertion cost		deletion cost		insertion cost	
	existence	extension	existence	extension	existence	extension	existence	extension
E_coli-K12	8	16	10	15	9	15	12	12
P_falciparum	12	8	8	12	13	7	10	10
R_sphaeroides	8	16	10	15	8	15	13	12

The gap cost for the 1st gap is equal to existence + extension. local and overlap alignments

Table S18. Alignment result for ION PGM datasets

Reference	local alignments					overlap alignments				
	aligned rate		substitution rate			aligned rate		substitution rate		
	seq	base	mismatch	ins	del	seq	base	mismatch	ins	del
E_coli-K12	98.5	97.8	0.0	0.4	0.5	98.0	98.0	0.1	0.5	0.6
P_falciparum	94.8	90.8	0.1	1.3	1.5	89.8	89.1	0.2	1.6	1.7
R_sphaeroides	98.6	97.9	0.0	0.4	0.6	97.9	97.8	0.0	0.5	0.7

The 3rd and 4th (8th and 9th) columns show aligned rates for sequence level and base level, respectively.

Ecoli_R73-2				Ecoli_R73-f				Ecoli_R73-hq-2				Ecoli_R73-hq-f							
A	C	G	T	A	C	G	T	A	C	G	T	A	C	G	T				
A	6	-12	-10	-22	A	5	-3	-1	-9	A	6	-12	-10	-22	A	5	-4	-2	-9
C	-9	6	-9	-10	C	-3	5	-4	-8	C	-10	6	-9	-10	C	-3	5	-5	-9
G	-8	-8	6	-10	G	-2	-4	5	-7	G	-8	-8	6	-10	G	-2	-4	5	-8
T	-21	-10	-12	6	T	-12	-9	-8	6	T	-21	-11	-12	6	T	-13	-9	-9	6
Ecoli_R73-hq-r				Ecoli_R73-r				ERR732541				ERR732542							
A	C	G	T	A	C	G	T	A	C	G	T	A	C	G	T				
A	5	-3	-3	-8	A	5	-3	-3	-8	A	5	-5	-5	-14	A	4	-1	-2	-10
C	-3	5	-5	-10	C	-3	5	-5	-10	C	-8	6	-5	-7	C	-3	5	-3	-8
G	-3	-5	5	-8	G	-3	-5	5	-8	G	-6	-4	6	-8	G	-2	-4	5	-9
T	-13	-11	-9	6	T	-13	-11	-9	6	T	-14	-6	-4	5	T	-10	-7	-5	5
ERR732543				ERR732544				ERR732545				ERR732546							
N/A				A	C	G	T	A	C	G	T	A	C	G	T				
				A	6	-9	-10	-27	A	4	-2	-3	-11	A	4	-2	-2	-11	
				C	-14	7	-8	-11	C	-4	5	-4	-11	C	-4	5	-4	-9	
				G	-12	-8	7	-13	G	-3	-5	5	-11	G	-2	-4	5	-9	
				T	-25	-9	-8	5	T	-11	-9	-7	5	T	-10	-8	-5	5	
ERR732547				ERR732548				ERR732549				ERR732550							
A	C	G	T	A	C	G	T	N/A				A	C	G	T				
A	5	-5	-5	-14	A	4	-1	-2	-9					A	6	-9	-10	-25	
C	-8	6	-5	-7	C	-3	5	-3	-8					C	-13	6	-8	-11	
G	-6	-4	6	-7	G	-2	-3	5	-8					G	-11	-7	6	-12	
T	-14	-6	-4	5	T	-9	-7	-5	5					T	-23	-9	-7	5	
ERR732551				ERR732552				ERR732553				ERR732554							
A	C	G	T	A	C	G	T	A	C	G	T	A	C	G	T				
A	4	-2	-3	-11	A	4	-2	-2	-11	A	5	-5	-4	-14	A	4	-1	-2	-9
C	-4	5	-4	-10	C	-4	5	-4	-9	C	-7	6	-5	-7	C	-3	5	-3	-8
G	-3	-5	5	-10	G	-2	-4	5	-9	G	-6	-4	6	-8	G	-2	-4	5	-8
T	-11	-9	-6	5	T	-10	-8	-5	5	T	-14	-6	-5	5	T	-9	-7	-5	5
ERR732555				ERR732556				ERR732557				ERR732558							
A	C	G	T	A	C	G	T	A	C	G	T	A	C	G	T				
A	4	-2	-2	-10	A	6	-10	-10	-28	A	4	-2	-3	-11	A	4	-2	-2	-11
C	-3	5	-4	-8	C	-14	7	-9	-12	C	-4	5	-4	-11	C	-4	5	-4	-9
G	-2	-3	5	-9	G	-12	-8	7	-13	G	-3	-5	5	-10	G	-2	-4	5	-10
T	-10	-8	-6	5	T	-27	-10	-9	5	T	-11	-8	-7	5	T	-11	-8	-6	5
H566_30_min_inc-2				H566_30_min_inc-f				H566_30_min_inc-hq-2				H566_30_min_inc-hq-f							
A	C	G	T	A	C	G	T	A	C	G	T	A	C	G	T				
A	6	-9	-7	-18	A	5	-3	-2	-10	A	6	-10	-8	-25	A	5	-4	-2	-10
C	-8	6	-8	-10	C	-3	5	-4	-8	C	-9	6	-8	-10	C	-3	5	-4	-8
G	-7	-7	6	-9	G	-2	-5	4	-7	G	-8	-8	6	-10	G	-2	-5	5	-8
T	-19	-9	-11	6	T	-12	-9	-8	6	T	-24	-10	-11	6	T	-12	-9	-9	6
H566_30_min_inc-hq-r				H566_30_min_inc-r				H566_ON_inc-2				H566_ON_inc-f							
A	C	G	T	A	C	G	T	A	C	G	T	A	C	G	T				
A	5	-3	-2	-9	A	5	-3	-2	-8	A	6	-9	-7	-19	A	5	-4	-2	-10
C	-2	5	-4	-8	C	-2	5	-4	-8	C	-8	6	-8	-10	C	-3	5	-5	-8
G	-2	-4	4	-7	G	-3	-5	4	-7	G	-7	-7	6	-9	G	-2	-5	5	-8
T	-11	-8	-8	6	T	-10	-8	-7	6	T	-20	-10	-11	6	T	-12	-9	-9	6
H566_ON_inc-hq-2				H566_ON_inc-hq-f				H566_ON_inc-hq-r				H566_ON_inc-r							
A	C	G	T	N/A				A	C	G	T	A	C	G	T				
A	6	-12	-9	-24					A	5	-3	-2	-10	A	5	-3	-2	-9	
C	-10	6	-9	-11					C	-2	5	-4	-9	C	-2	5	-4	-8	
G	-9	-8	6	-9					G	-3	-5	5	-8	G	-3	-5	4	-7	
T	-23	-10	-11	6					T	-11	-9	-9	6	T	-11	-8	-8	6	

Fig. S8. Substitution score matrix for Nanopore (local alignment). N/A indicates LAST-TRAIN failed for training.

Ecoli_R73-2				Ecoli_R73-f				Ecoli_R73-hq-2				Ecoli_R73-hq-f							
A	C	G	T	A	C	G	T	A	C	G	T	A	C	G	T				
A	6	-14	-11	-24	A	5	-4	-2	-9	A	6	-14	-11	-24	A	5	-4	-2	-9
C	-11	6	-10	-11	C	-3	5	-5	-9	C	-11	6	-10	-11	C	-3	5	-5	-9
G	-9	-9	6	-11	G	-2	-4	5	-8	G	-9	-9	6	-11	G	-2	-4	5	-8
T	-23	-12	-13	6	T	-13	-9	-9	6	T	-23	-12	-13	6	T	-13	-9	-9	6
Ecoli_R73-hq-r				Ecoli_R73-r				ERR732541				ERR732542							
A	C	G	T	A	C	G	T	A	C	G	T	A	C	G	T				
A	5	-4	-3	-9	A	5	-4	-3	-9	A	5	-6	-7	-17	A	4	-2	-2	-10
C	-3	5	-5	-10	C	-3	5	-5	-10	C	-12	6	-7	-8	C	-4	5	-4	-9
G	-3	-5	5	-8	G	-3	-5	5	-8	G	-10	-5	6	-9	G	-3	-5	5	-10
T	-13	-11	-9	6	T	-13	-11	-9	6	T	-23	-9	-7	6	T	-11	-8	-7	6
ERR732543				ERR732544				ERR732545				ERR732546							
A	C	G	T	A	C	G	T	A	C	G	T	A	C	G	T				
A	4	-2	-2	-10	A	6	-11	-11	-28	A	4	-2	-3	-11	A	4	-2	-2	-11
C	-4	5	-4	-8	C	-15	7	-9	-12	C	-4	6	-4	-10	C	-5	5	-4	-9
G	-2	-4	5	-9	G	-13	-8	7	-14	G	-3	-6	5	-11	G	-2	-4	5	-9
T	-11	-8	-6	5	T	-26	-11	-9	6	T	-12	-9	-7	6	T	-12	-9	-6	5
ERR732547				ERR732548				ERR732549				ERR732550							
A	C	G	T	A	C	G	T	A	C	G	T	A	C	G	T				
A	5	-6	-7	-16	A	4	-2	-2	-10	A	4	-2	-2	-10	A	6	-11	-11	-25
C	-11	6	-6	-8	C	-4	5	-4	-8	C	-4	5	-4	-8	C	-14	7	-9	-12
G	-9	-5	6	-8	G	-2	-4	5	-9	G	-2	-4	5	-8	G	-12	-8	7	-13
T	-20	-8	-7	6	T	-11	-8	-6	5	T	-11	-8	-6	5	T	-25	-11	-9	6
ERR732551				ERR732552				ERR732553				ERR732554							
A	C	G	T	A	C	G	T	A	C	G	T	A	C	G	T				
A	4	-2	-3	-10	A	4	-2	-2	-11	A	5	-6	-7	-17	A	4	-2	-2	-10
C	-4	6	-4	-10	C	-5	5	-4	-9	C	-10	6	-6	-9	C	-3	5	-4	-9
G	-3	-6	5	-10	G	-2	-4	5	-9	G	-9	-5	6	-9	G	-3	-4	5	-9
T	-12	-9	-7	6	T	-12	-9	-6	5	T	-21	-9	-7	6	T	-11	-8	-7	5
ERR732555				ERR732556				ERR732557				ERR732558							
A	C	G	T	A	C	G	T	A	C	G	T	A	C	G	T				
A	4	-2	-2	-11	A	6	-11	-11	-28	A	4	-2	-3	-11	A	4	-2	-2	-11
C	-4	5	-4	-9	C	-15	7	-9	-13	C	-4	6	-4	-10	C	-4	5	-4	-9
G	-2	-4	5	-9	G	-12	-8	7	-14	G	-3	-6	5	-10	G	-2	-4	5	-10
T	-11	-8	-6	5	T	-28	-11	-10	6	T	-12	-9	-7	6	T	-12	-9	-6	5
H566_30_min_inc-2				H566_30_min_inc-f				H566_30_min_inc-hq-2				H566_30_min_inc-hq-f							
A	C	G	T	A	C	G	T	A	C	G	T	A	C	G	T				
A	6	-10	-8	-20	A	5	-4	-2	-10	A	6	-12	-9	-27	A	5	-4	-2	-10
C	-9	6	-9	-11	C	-3	5	-5	-8	C	-10	6	-9	-11	C	-3	5	-5	-8
G	-8	-8	6	-10	G	-2	-5	5	-8	G	-9	-8	6	-10	G	-2	-5	5	-8
T	-21	-11	-12	6	T	-12	-9	-9	6	T	-25	-11	-13	6	T	-13	-9	-9	6
H566_30_min_inc-hq-r				H566_30_min_inc-r				H566_ON_inc-2				H566_ON_inc-f							
A	C	G	T	A	C	G	T	A	C	G	T	A	C	G	T				
A	5	-3	-2	-10	A	5	-3	-2	-9	A	6	-10	-8	-20	A	5	-4	-2	-10
C	-3	5	-4	-9	C	-2	5	-4	-8	C	-9	6	-9	-11	C	-3	5	-5	-8
G	-2	-5	5	-8	G	-3	-5	4	-8	G	-7	-8	6	-10	G	-2	-5	5	-8
T	-11	-8	-9	6	T	-11	-8	-8	6	T	-21	-11	-12	6	T	-12	-9	-9	6
H566_ON_inc-hq-2				H566_ON_inc-hq-f				H566_ON_inc-hq-r				H566_ON_inc-r							
A	C	G	T	A	C	G	T	A	C	G	T	A	C	G	T				
A	6	-14	-11	-24	A	5	-4	-2	-10	A	5	-3	-3	-10	A	5	-3	-2	-9
C	-11	6	-10	-12	C	-3	5	-4	-8	C	-3	5	-5	-10	C	-2	5	-4	-8
G	-10	-10	6	-11	G	-2	-5	5	-8	G	-3	-6	5	-9	G	-3	-5	5	-8
T	-23	-12	-13	6	T	-12	-9	-9	6	T	-12	-10	-9	6	T	-11	-9	-8	6

Fig. S9. Substitution score matrix for Nanopore (overlap alignment) N/A indicates LAST-TRAIN failed for training.

Table S19. Trained affine gap costs for Oxford Nanopore sequencers

Dataset	local alignment				overlap alignment			
	deletion cost		insertion cost		deletion cost		insertion cost	
	existence	extension	existence	extension	existence	extension	existence	extension
Ecoli_R73-2	11	3	14	3	11	4	14	3
Ecoli_R73-f	9	4	17	3	9	4	16	3
Ecoli_R73-hq-2	11	4	14	3	11	4	14	3
Ecoli_R73-hq-f	9	4	17	3	9	4	17	3
Ecoli_R73-hq-r	10	4	14	3	9	4	14	3
Ecoli_R73-r	10	4	14	3	9	4	14	3
ERR732541	10	4	15	3	10	4	14	3
ERR732542	10	4	16	2	9	4	15	3
ERR732543	-	-	-	-	8	4	18	3
ERR732544	10	5	14	4	10	5	14	4
ERR732545	9	4	15	3	9	4	15	3
ERR732546	8	4	18	3	8	4	17	3
ERR732547	10	4	15	3	10	4	14	3
ERR732548	10	3	17	2	9	4	16	2
ERR732549	-	-	-	-	8	4	19	3
ERR732550	10	4	14	4	10	5	15	3
ERR732551	9	4	16	3	9	4	15	3
ERR732552	8	4	19	3	8	4	18	3
ERR732553	11	3	15	3	10	4	14	3
ERR732554	11	3	16	3	10	4	15	3
ERR732555	9	4	18	3	8	4	17	3
ERR732556	10	4	14	4	11	4	15	3
ERR732557	9	4	15	3	9	4	15	3
ERR732558	8	4	17	3	8	4	17	3
H566_30_min_inc-2	12	3	14	3	12	3	14	3
H566_30_min_inc-f	9	4	17	2	9	4	16	3
H566_30_min_inc-hq-2	12	3	14	3	12	4	14	3
H566_30_min_inc-hq-f	9	4	16	3	9	4	16	3
H566_30_min_inc-hq-r	10	3	16	2	10	4	16	3
H566_30_min_inc-r	13	2	17	3	13	2	17	3
H566_ON_inc-2	12	3	14	3	12	3	14	3
H566_ON_inc-f	9	4	17	3	9	4	16	3
H566_ON_inc-hq-2	12	4	14	3	11	4	14	3
H566_ON_inc-hq-f	-	-	-	-	9	4	17	3
H566_ON_inc-hq-r	10	4	15	2	9	4	15	2
H566_ON_inc-r	13	2	17	3	12	2	17	3

The gap cost for the 1st gap is equal to existence + extension. local and overlap alignments

Table S20. Alignment results for Oxford Nanopore sequencers

Reference	local alignment					overlap alignment				
	aligned rate		substitution rate			aligned rate		substitution rate		
	seq	base	mismatch	ins	del	seq	base	mismatch	ins	del
Ecoli_R73-2	75.5	65.5	5.7	6.6	10.7	29.0	26.3	5.0	6.8	8.7
Ecoli_R73-f	51.0	52.3	21.1	3.3	14.6	29.1	28.6	19.6	3.8	13.9
Ecoli_R73-hq-2	86.2	66.9	5.8	6.2	9.8	34.9	30.0	5.0	6.8	8.6
Ecoli_R73-hq-f	72.2	72.5	19.7	3.4	13.9	48.9	50.6	19.6	3.5	13.6
Ecoli_R73-hq-r	69.7	64.7	17.6	5.1	12.7	41.7	42.3	16.9	5.7	12.2
Ecoli_R73-r	60.4	58.6	17.7	5.3	12.6	34.5	37.3	17.0	5.9	12.2
ERR732541	59.2	22.9	12.8	5.1	10.1	13.0	3.3	9.5	6.6	10.6
ERR732542	-	-	-	-	-	22.7	14.7	19.4	5.8	11.7
ERR732543	-	-	-	-	-	20.7	12.2	19.7	2.1	15.3
ERR732544	73.7	52.5	5.7	4.6	8.3	52.2	26.4	5.2	5.0	8.2
ERR732545	72.3	41.3	17.1	4.9	11.8	55.7	34.9	17.6	5.4	11.8
ERR732546	69.3	41.1	19.5	1.9	15.5	40.5	16.2	18.8	2.4	14.8
ERR732547	56.3	19.2	13.0	5.1	10.6	11.0	2.8	10.4	6.5	11.0
ERR732548	-	-	-	-	-	-	-	-	-	-
ERR732549	-	-	-	-	-	16.1	6.6	19.8	2.2	15.6
ERR732550	72.2	39.9	5.8	4.9	8.4	46.7	20.2	5.4	5.9	8.3
ERR732551	70.6	34.6	17.4	4.6	12.0	46.9	26.5	17.9	5.5	12.0
ERR732552	65.5	32.8	19.2	2.0	15.4	32.5	11.6	18.6	2.8	14.2
ERR732553	57.8	24.0	13.2	5.6	10.8	10.8	3.0	9.7	6.9	10.8
ERR732554	53.8	36.2	21.7	4.5	13.1	11.1	5.3	19.7	5.5	11.0
ERR732555	51.3	34.3	20.0	2.1	14.9	24.2	14.2	19.3	2.5	14.9
ERR732556	76.1	52.1	4.9	5.0	8.4	53.8	28.5	4.6	5.8	8.4
ERR732557	74.4	38.7	17.1	4.7	12.1	51.9	29.4	17.7	5.4	12.3
ERR732558	73.4	45.5	18.7	2.2	15.1	50.8	21.2	18.7	2.5	14.8
H566_30_min_inc-2	87.9	63.6	7.7	8.2	11.4	34.4	29.2	6.5	7.8	11.1
H566_30_min_inc-f	-	-	-	-	-	35.7	29.9	19.2	5.4	12.1
H566_30_min_inc-hq-2	70.7	50.9	6.4	7.4	9.8	23.9	25.6	5.9	7.7	8.1
H566_30_min_inc-hq-f	52.1	65.3	19.5	5.6	11.7	37.7	51.9	19.1	5.4	11.8
H566_30_min_inc-hq-r	50.8	48.1	20.6	6.1	13.8	17.4	16.9	20.8	5.4	12.2
H566_30_min_inc-r	75.3	63.8	19.1	3.1	21.2	37.0	34.1	18.6	3.0	21.7
H566_ON_inc-2	87.6	70.5	7.6	8.4	10.7	39.0	35.6	6.7	8.6	10.4
H566_ON_inc-f	62.2	52.2	19.3	4.6	12.8	47.2	43.2	18.9	4.5	13.0
H566_ON_inc-hq-2	70.2	55.3	5.9	9.1	7.8	37.1	31.4	4.7	8.4	7.6
H566_ON_inc-hq-f	-	-	-	-	-	50.3	48.9	19.5	4.7	12.9
H566_ON_inc-hq-r	-	-	-	-	-	-	-	-	-	-
H566_ON_inc-r	74.9	62.4	18.1	2.7	22.6	61.7	64.1	17.5	2.8	23.3

See Table S5 for the details of each dataset. The 3rd and 4th (8th and 9th) columns show aligned rates for sequence level and base level, respectively.

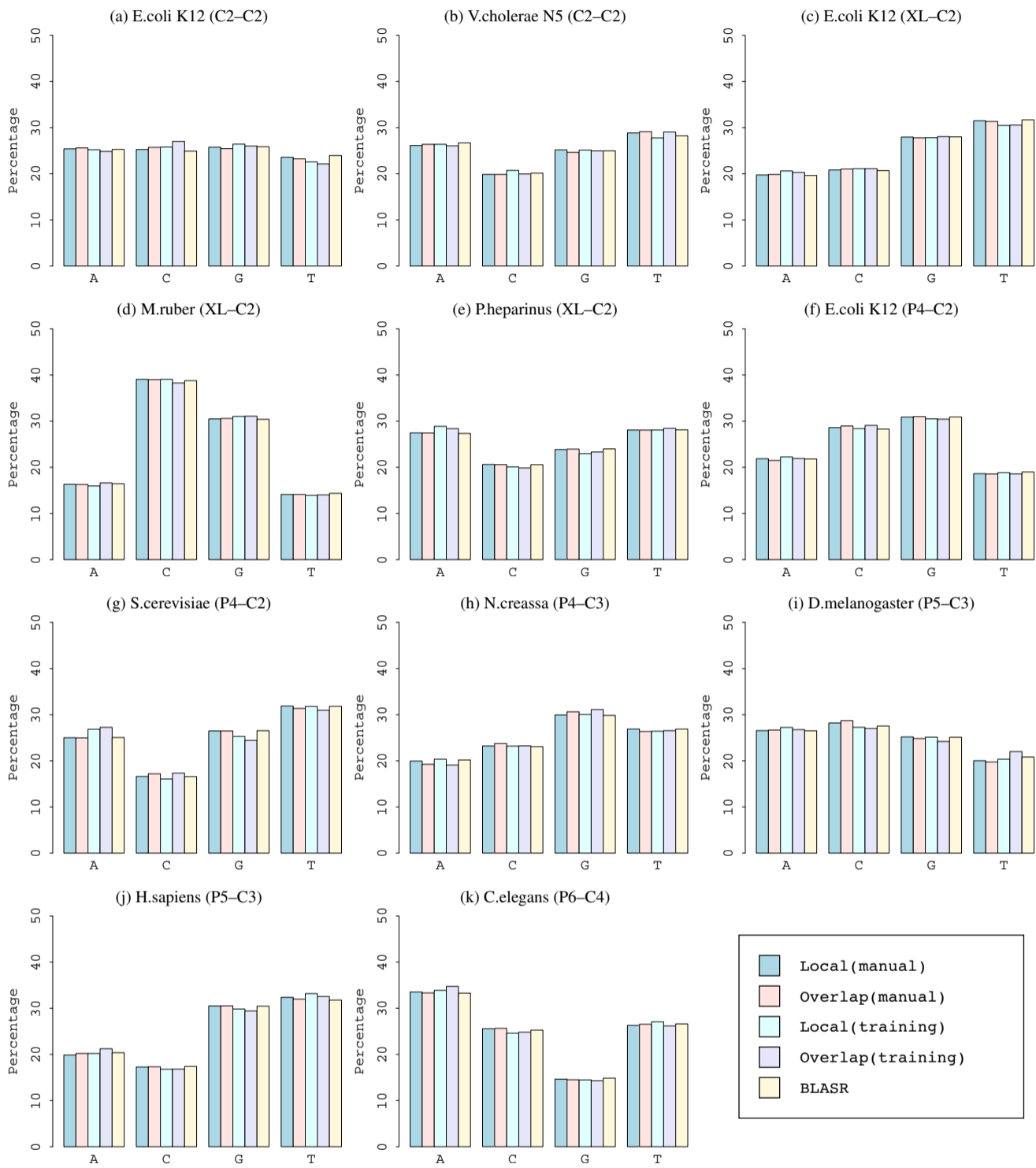


Fig. S10. The distribution of nucleotide type of deletions for PacBio RS.

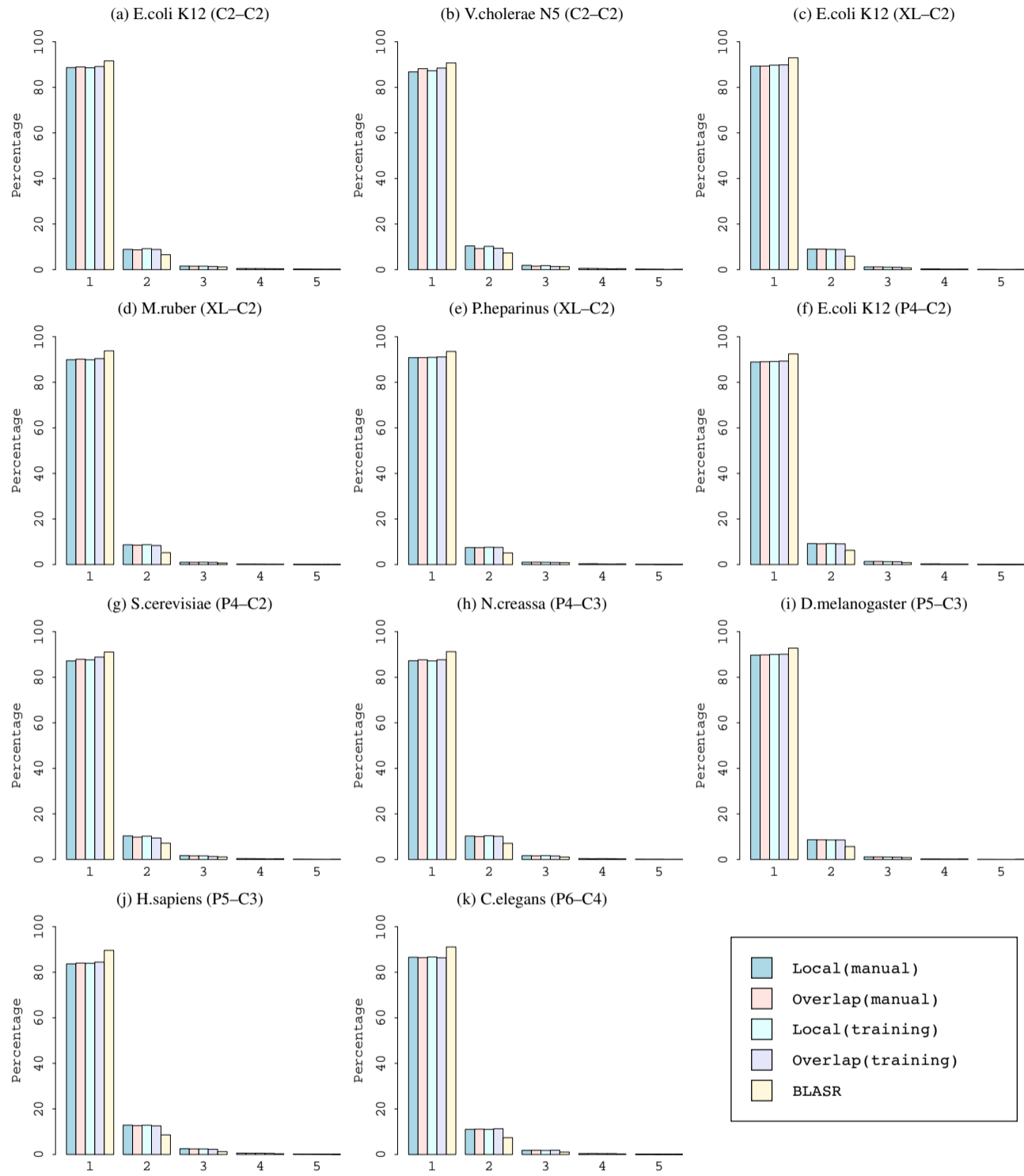


Fig. S11. The size of deletions for PacBio RS.

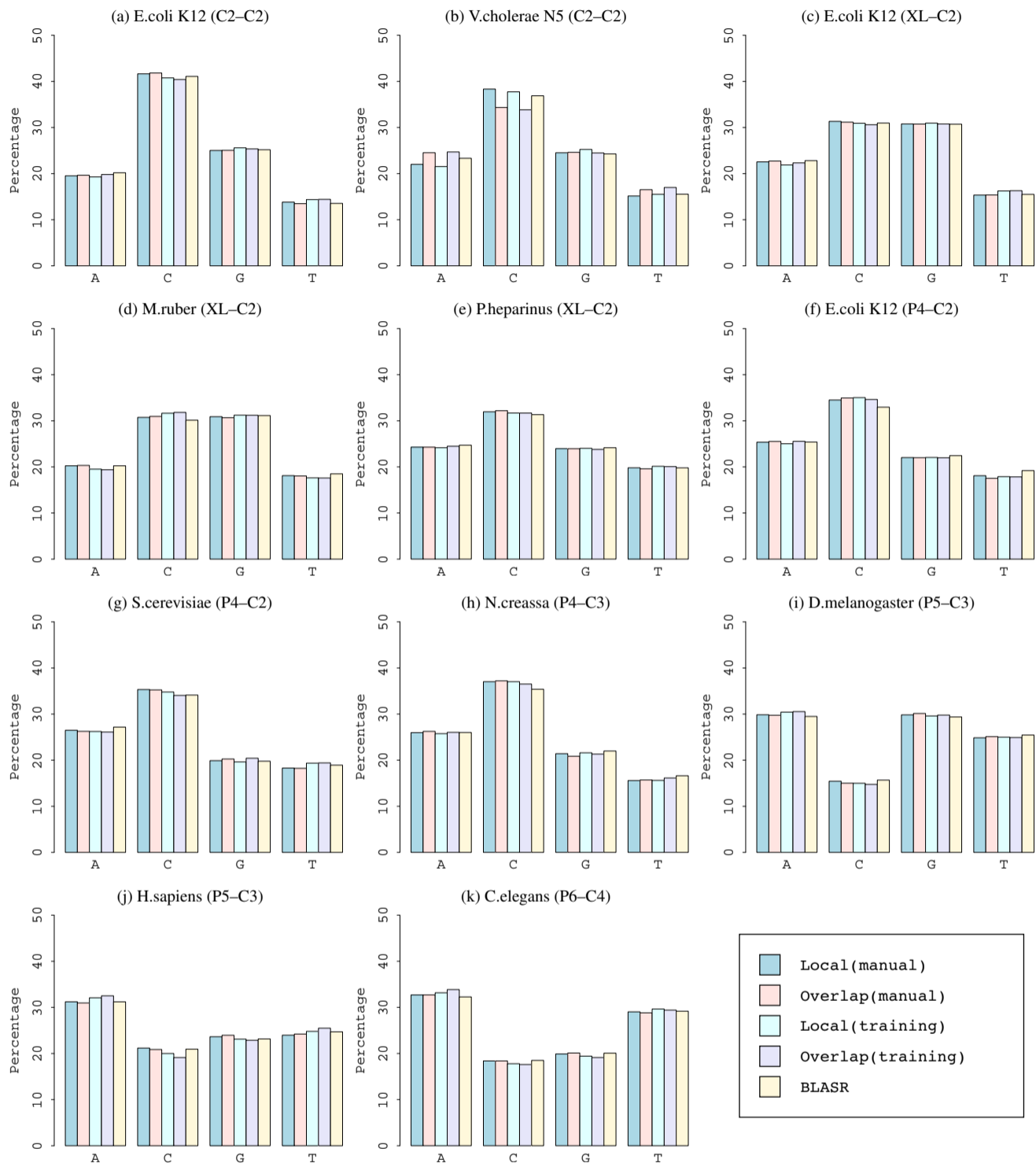


Fig. S12. The distribution of nucleotide type of insertion for PacBio RS.

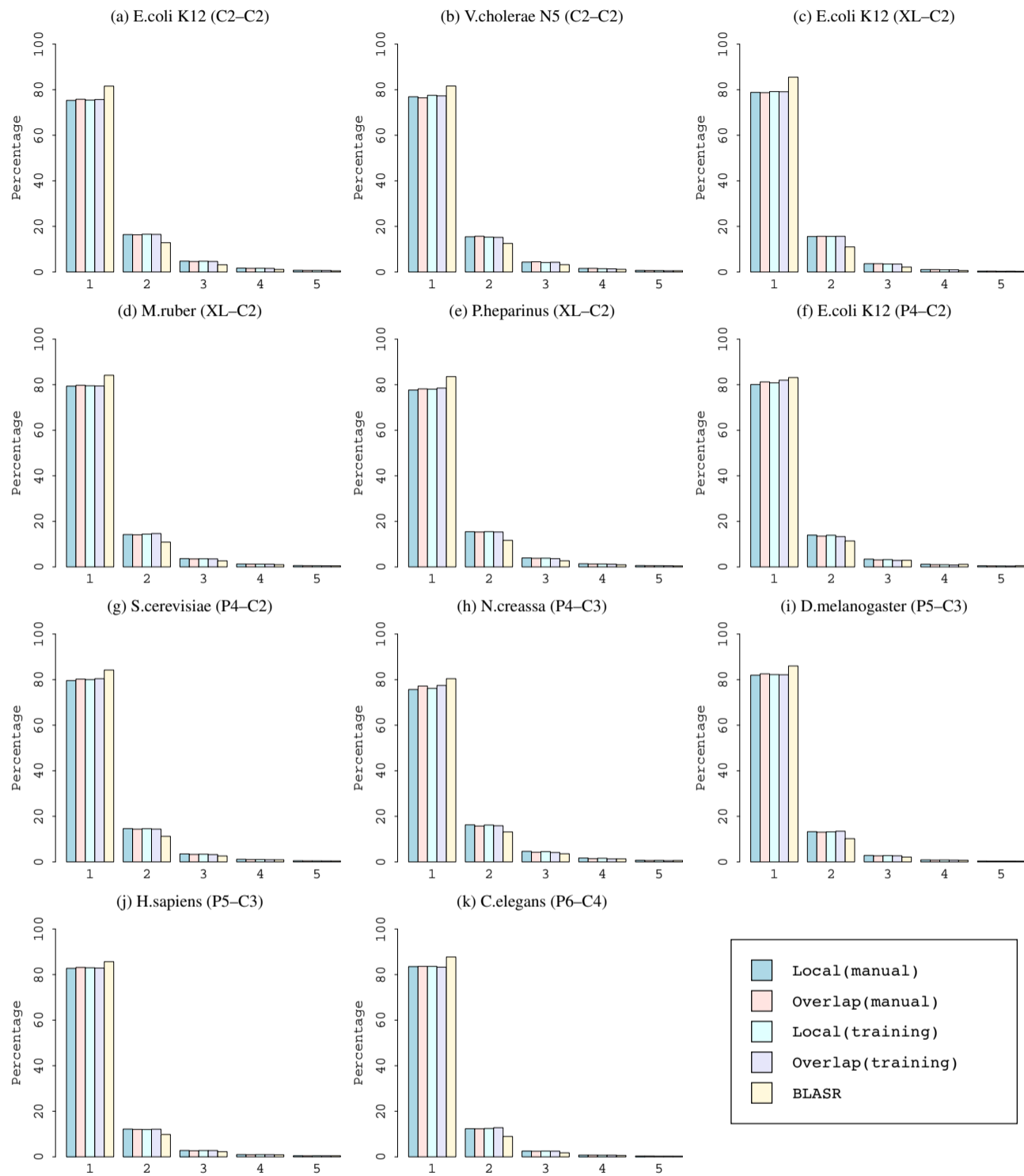


Fig. S13. The size of insertions for PacBio RS.

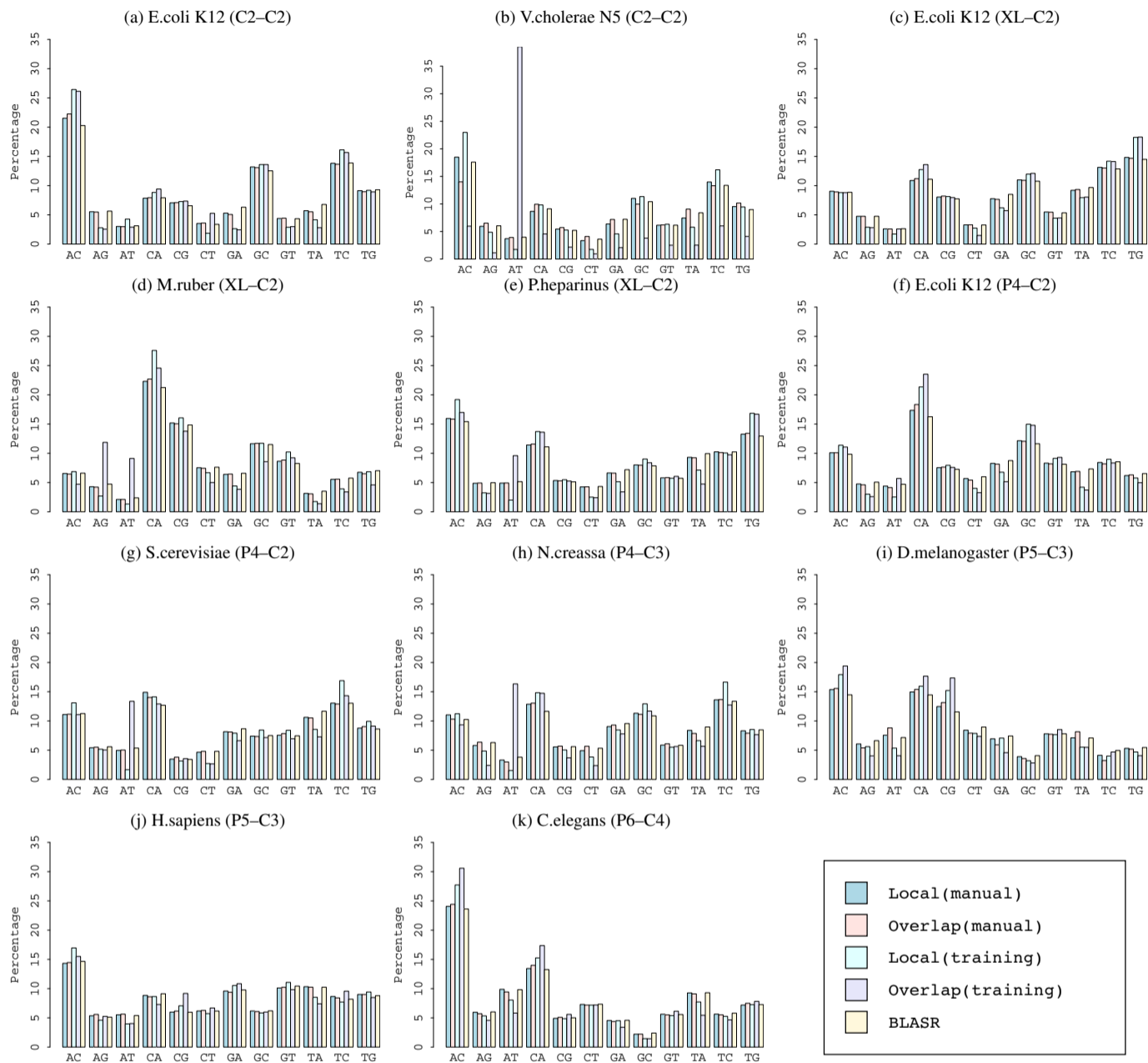


Fig. S14. Frequency of patterns of mis-match (substitution) nucleotides for PacBio RS.

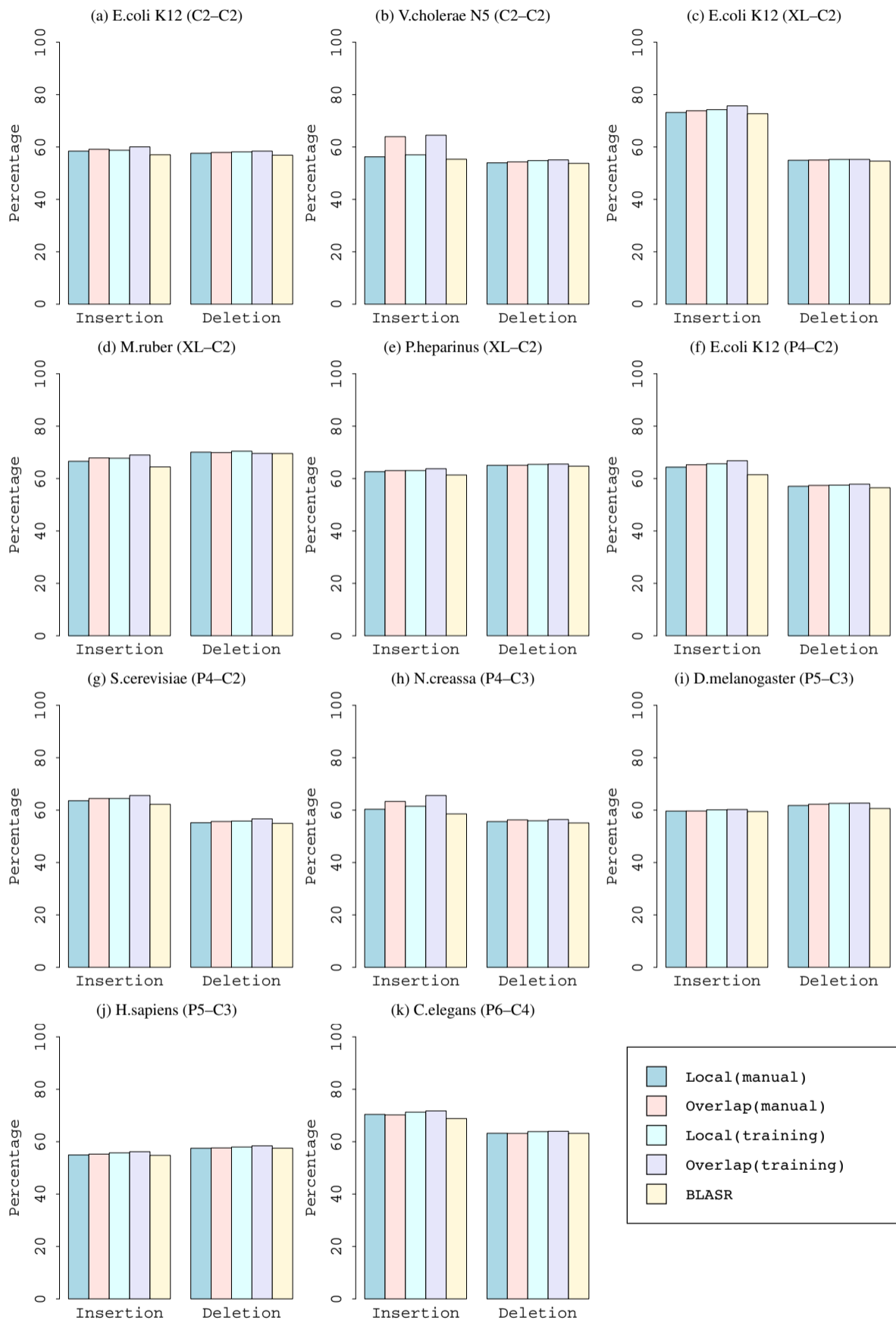


Fig. S15. Frequency of neighboring nucleotides that are identical with the deleted/inserted nucleotide for PacBio RS.

References

- Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**(17), 3389–3402.
- Ammar, R., Paton, T. A., Torti, D., Shlien, A., and Bader, G. D. (2015). Long read nanopore sequencing for detection of HLA and CYP2D6 variants and haplotypes. *F1000Res*, **4**, 17.
- Ashton, P. M., Nair, S., Dallman, T., Rubino, S., Rabsch, W., Mwaigwisya, S., Wain, J., and O’Grady, J. (2015). MinION nanopore sequencing identifies the position and structure of a bacterial antibiotic resistance island. *Nat. Biotechnol.*, **33**(3), 296–300.
- Bragg, L. M., Stone, G., Butler, M. K., Hugenholtz, P., and Tyson, G. W. (2013). Shining a light on dark sequencing: characterising errors in Ion Torrent PGM data. *PLoS Comput. Biol.*, **9**(4), e1003031.
- Chaisson, M. J. and Tesler, G. (2012). Mapping single molecule sequencing reads using basic local alignment with successive refinement (BLASR): application and theory. *BMC Bioinformatics*, **13**, 238.
- Chiaromonte, F., Yap, V. B., and Miller, W. (2002). Scoring pairwise genomic sequence alignments. *Pac Symp Biocomput*, pages 115–126.
- Chin, C. S., Alexander, D. H., Marks, P., Klammer, A. A., Drake, J., Heiner, C., Clum, A., Copeland, A., Huddleston, J., Eichler, E. E., Turner, S. W., and Korlach, J. (2013). Nonhybrid, finished microbial genome assemblies from long-read SMRT sequencing data. *Nat. Methods*, **10**(6), 563–569.
- Durbin, R., Eddy, S., Krogh, A., and Mitchison, G. (1998). *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge University Press.
- Frith, M. and Kawaguchi, R. (2015). Split-alignment of genomes finds orthologies more accurately. *Genome biology*, **16**(1), 106–106.
- Frith, M. C., Park, Y., Sheetlin, S. L., and Spouge, J. L. (2008). The whole alignment and nothing but the alignment: the problem of spurious alignment flanks. *Nucleic Acids Res.*, **36**(18), 5863–5871.
- Frith, M. C., Wan, R., and Horton, P. (2010a). Incorporating sequence quality data into alignment improves DNA read mapping. *Nucleic Acids Res.*, **38**, e100.
- Frith, M. C., Hamada, M., and Horton, P. (2010b). Parameters for accurate genome alignment. *BMC Bioinformatics*, **11**, 80.
- Hamada, M., Wijaya, E., Frith, M. C., and Asai, K. (2011). Probabilistic alignments with quality scores: an application to short-read mapping toward accurate SNP/indel detection. *Bioinformatics*, **27**(22), 3085–3092.
- Jain, M., Fiddes, I. T., Miga, K. H., Olsen, H. E., Paten, B., and Akeson, M. (2015). Improved data analysis for the MinION nanopore sequencer. *Nat. Methods*, **12**(4), 351–356.
- Junemann, S., Sedlazeck, F. J., Prior, K., Albersmeier, A., John, U., Kalinowski, J., Mellmann, A., Goesmann, A., von Haeseler, A., Stoye, J., and Harmsen, D. (2013). Updating benchtop sequencing performance comparison. *Nat. Biotechnol.*, **31**(4), 294–296.
- Kerpedjiev, P., Frellsen, J., Lindgreen, S., and Krogh, A. (2014). Adaptable probabilistic mapping of short reads using position specific scoring matrices. *BMC Bioinformatics*, **15**, 100.
- Laszlo, A. H. and Derrington, I. M. *et al.* (2014). Decoding long nanopore sequencing reads of natural DNA. *Nat. Biotechnol.*, **32**(8), 829–833.
- Laver, T. W., Caswell, R. C., Moore, K. A., Poschmann, J., Johnson, M. B., Owens, M. M., Ellard, S., Paszkiewicz, K. H., and Weedon, M. N. (2016). Pitfalls of haplotype phasing from amplicon-based long-read sequencing. *Sci Rep*, **6**, 21746.
- Loman, N. J., Quick, J., and Simpson, J. T. (2015). A complete bacterial genome assembled de novo using only nanopore sequencing data. *Nat. Methods*, **12**(8), 733–735.
- Miyazawa, S. (1995). A reliable sequence alignment method based on probabilities of residue correspondences. *Protein Eng.*, **8**, 999–1009.
- Numanagi, I., Maliki, S., Pratt, V. M., Skaar, T. C., Flockhart, D. A., and Sahinalp, S. C. (2015). Cypiripi: exact genotyping of CYP2D6 using high-throughput sequencing data. *Bioinformatics*, **31**(12), 27–34.
- Ono, Y., Asai, K., and Hamada, M. (2013). PBSIM: PacBio reads simulator—toward accurate genome assembly. *Bioinformatics*, **29**(1), 119–121.
- Quick, J., Quinlan, A. R., and Loman, N. J. (2014). A reference bacterial genome dataset generated on the MinION portable single-molecule nanopore sequencer. *Gigascience*, **3**, 22.
- Ross, M. G., Russ, C., Costello, M., Hollinger, A., Lennon, N. J., Hegarty, R., Nusbaum, C., and Jaffe, D. B. (2013). Characterizing and measuring bias in sequence data. *Genome Biol.*, **14**(5), R51.
- Schwartz, A. S., Myers, E. W., and Pachter, L. (2005). Alignment Metric Accuracy. *arXiv:q-bio/0510052*.
- Sheetlin, S., Park, Y., Frith, M. C., and Spouge, J. L. (2016). ALP & FALP: C++ libraries for pairwise local alignment E-values. *Bioinformatics*, **32**(2), 304–305.
- Sovic, I., Sikic, M., Wilm, A., Fenlon, S. N., Chen, S., and Nagarajan, N. (2016). Fast and sensitive mapping of nanopore sequencing reads with GraphMap. *Nat Commun*, **7**, 11307.
- States, D. J., Gish, W., and Altschul, S. F. (1991). Improved sensitivity of nucleic acid database similarity searches using application specific scoring matrices. *Methods: A companion to Methods in Enzymology*, **3**, 66–70.
- Twist, G. P., Gaedigk, A., Miller, N. A., Farrow, E. G., Willig, L. K., Dinwiddie, D. L., Petrikov, J. E., Soden, S. E., Herd, S., Gibson, M., Cakici, J. A., Riffel, A. K., Leeder, J. S., Dinakarandian, D., and Kingsmore, S. F. (2016). Constellation: a tool for rapid, automated phenotype assignment of a highly polymorphic pharmacogene, CYP2D6, from whole-genome sequences. *npj Genomic Medicine*, **1**, 15007+.
- Yu, Y. K. and Altschul, S. F. (2005). The construction of amino acid substitution matrices for the comparison of proteins with non-standard compositions. *Bioinformatics*, **21**(7), 902–911.
- Zhang, Z., Berman, P., and Miller, W. (1998). Alignments without low-scoring regions. *J. Comput. Biol.*, **5**(2), 197–210.