

Supplementary Text and Figures

Availability of code and data

All code and configuration files used for the analyses in this manuscript are available at https://github.com/valenlab/amplican_manuscript where they can be used for independent verification. Data analysed is publicly available and accessible at their respective accession numbers. `amplican` is available from Bioconductor as the R package at <http://bioconductor.org/packages/amplican>. To obtain the newest development version visit <https://github.com/valenlab/amplican>.

Supplementary Note 1: Automatic normalization using control reads

By default `amplican` normalizes through the strict removal of all indel events also found in the control sample above a threshold (default 1%). Due to the stochasticity in sequencing data this approach is better suited to handle more heterogeneous cases than the subtraction method where indel frequency is simply normalized by subtraction of the control indel frequency. In the latter case variation in the levels of indel frequencies in the control versus CRISPR treated samples can lead to partially normalized data. Both normalization methods are outlined in **Supplementary Fig. 16** while **Supplementary Fig. 17** shows examples of highly heterogeneous data where without normalization estimated mutation efficiency would be biased. Normalization becomes even more important when the exact nature of the indel event matters, for instance whether it induces a frameshift or not.

All examples below and all other experiments that were not presented here can be recreated with the use of “`make_comparison_normalized.Rmd`” in the `amplican_manuscript` repository.

Supplementary Note 2: `amplican` utilizes optimized alignments

CRISPR genome editing events typically result in a single break at a single site and by extension produce a single deletion and/or insertion. Sequence read aligners are generally not optimized for this type of genome editing event which can lead to the aligner fragmenting the indels and creating multiple events (example in **Supplementary Fig. 2**). In the worst case fragmented alignments could shift the indel events resulting in a distortion of the mutation efficiency for those tools that only allow events within a certain distance from the expected site. A more likely outcome however, is the misinterpretation of the nature of the mutation.

Under certain assumptions the theoretically optimal alignment can be obtained by the Needleman-Wunsch algorithm. `amplican` uses this algorithm with optimized parameters to reflect the expectation that a CRISPR experiment should result in one deletion and/or insertion event, of unknown length (match = 5, mismatch = -4, gap opening = 25, gap extension = 0, no end gap penalties). With these parameters the Needleman-Wunsch algorithm performs well over a broad range of test cases (data not shown). `amplican` uses these optimized parameters by default, but also allows for supervision of the alignments through human readable output of individual alignment results (**Supplementary Table 2**).

Supplementary Note 3: Synthetic data set evaluation

The latest available versions was used for all tools and packages. The assessment set from Lindsay et al.¹ paper (Synthetic Dataset 2, Supplementary 4) was replicated with the same settings and seed values as described (**Supplementary Fig. 3**). The script from Lindsay et al.

was used for parsing, but a bug in the code was fixed for the CRISPResso output. In Lindsay et al. only NHEJ estimation of mutation efficiency was considered for CRISPResso, skipping HDR and “mixed” mutation frequencies. However, fixing this error did not influence CRISPResso’s overall performance in any significant way. Versions of tools, scripts and details needed for replication are available in the https://github.com/valenlab/amplican_manuscript repository. amplican used the same amplicon sequences as CRISPResso.

It should be noted that Synthetic Dataset 2 from Lindsay et al. (used for **Supplementary Fig. 3**) is not a good approximation of a real life situation. First, the sequence matching the primers can not be very divergent as they would then fail to amplify. Second, several experiments are badly designed in that the target sites are very close to the sequencing end of the reads. This makes it difficult to correctly call indels with support from both paired reads. Third, paired-end sequencing of 200bp or longer is somewhat expensive and error-prone and most labs would seek to restrict this to shorter reads. To account for this we created an additional set, Synthetic Dataset 3, in a similar fashion to Synthetic Dataset 2, but with the following minor modifications. First, the length of amplicons and reads (150 bp) were adjusted. Second, gRNA target sites were designed to be covered by both reads. Third, PCR off-target reads were created without mutating the primer sequences. Finally, mutation efficiency was tested across a range of mismatch rates, 10%, 20% and 30% (**Fig. 2, Supplementary Fig. 4**), to reflect different levels of similarity to the contaminant reads.

For Synthetic Dataset 2 amplican matches the perfect score of CrispRVariants and AmpliconDivider. However, on Synthetic Dataset Dataset 3 amplican is more consistent at estimating the known mutation efficiencies within the dataset (**Fig. 2**). AmpliconDIVider has no filtering step and is confused by the contaminating reads. CrispRVariants has a filtering step, but is unable to discern divergent off-target sequences (e.g. homologous regions) that are still able to align to the correct target site. As in the benchmark from Lindsay et al. CRISPResso performs poorly on all benchmarks. When increasing mismatch rate (from 10% of all bases to 20% and 30%), AmpliconDIVider and CrispRVariants get closer to the correct estimated indel rate, but in all cases amplican obtains the highest precision and shows the most robust performance (**Fig. 2, Supplementary Fig. 4**).

Supplementary Note 4: amplican is able to correctly call longer indels

We have found that even without targeted insertion CRISPR mutagenesis can frequently result in some proportion of longer indels (**Supplementary Fig. 18**). In particular, we have observed unintended insertions from lentiviral vectors used to introduce the guides and Cas9 (**Supplementary Fig. 19**)².

Current tools primarily rely on either global mapping (CrispRVariants, AmpliconDIVider) (**Supplementary Tab 1**) that can have problems identifying the correct loci in the presence of a larger insertion or have certain processing steps that are incompatible with longer events (see below for CrispRVariants). This mitigates primer dimer contamination problems (which can be identified by too large deletion gaps after alignments), but ignores bona fide large indels. These tools are therefore often unable to handle longer indels whether unintended or targeted. Long deletions are also a problem for some tools. For instance, CrispRVariants filters out any deletion that does not start or end within the gRNA complementary sequence plus a buffer of 5 bp. Any deletion spanning this region is ignored. This can be used as a strategy to filter

primer-dimers, but also has the side-effect of ignoring any bona fide longer deletions. ampliCan uses a local alignment strategy that can detect these longer indels and a more realistic model of primer-dimer artifacts (**Supplementary Note 6**).

We noticed that in the Synthetic Dataset 2 from Lindsay et al. (CrisprVariants benchmark dataset) large indels (>10 bp) were disabled. To assess the capabilities of leading tools in handling longer indels we created Synthetic Dataset 4. We made three subsets: 1) with no indels > 10bp, 2) with a mix of indels by simply removing the line disabling longer reads in the Lindsay et al. script. 3) To check explicitly how tools handle experiments with planned shorter insertion of donor sequence we created a third scenario described as “insertions > 10bp” on the figure. 4) For completeness we also created a set of large deletions. ampliCan match the best competitors on the the set with no long indels and consistently outperform the other tools on the mixed set and the set only containing long indels (**Fig. 2, Supplementary Fig. 5**).

Supplementary Note 5: Visualization and aggregation of the complete activity of gRNAs

While the default alignment plot shows the most abundant reads across the expected cut site it doesn't provide an overview over all editing events. ampliCan therefore also produces multiple plots that aggregate and visualize editing events (**Supplementary Fig. 6**). Unlike the alignment plots these show the complete activity of the gRNA allowing for comparison of gRNAs by manual inspection. The pipeline in ampliCan treats forward and reverse reads separately which, after visualization, makes it possible to spot read-related problems immediately (**Supplementary Figs. 13, 14, 21**). In addition, ampliCan provides meta plots that aggregate information across groups of experiments allowing for visualization of deletions, mismatches or insertions across groups of gRNAs, amplicons or any other set. This can for instance show the combined activity of a single guide across multiple experiments.

ampliCan builds on top of ggbio³ and ggplot2⁴ packages and provides higher level functions that automatically group event data (eg. collapse on start and end of deletion) and plot results. Users can extend those plot objects and treat them like any other ggplot2 object. ampliCan supports multiple types of meta plots to facilitate comparison of not only the gRNAs, but also any group that a user wants e.g. barcode, amplicon, type of treatment (**Supplementary Figs. 9, 17, 20**).

Supplementary Note 6: Filtering of noise

Multiple sources of noise can confound the estimation of cut rates, low quality reads, primer-dimers, PCR off-target amplification and sequencing artifacts. ampliCan has three filters to remove noise from different sources: 1) low quality reads, 2) primer-dimers, and 3) erroneously assigned reads and sequencing artifacts.

1. Low quality reads

ampliCan offers basic read quality overview with the use of ShortRead⁵ package and filters for minimum base quality (default: 0) in a read, average base minimum quality (default avg min: 30) and the presence of ambiguous (N) letters.

2. Primer-dimers

Filtering primer-dimers is a balance between getting rid of erroneous reads and allowing for longer deletions. For instance, CrispRVariants ignores all alignments with deletions larger than 33 bp (the guide plus a buffer of 5 bp) and is frequently unable to map long insertions (**Fig. 2B**, **Supplementary Fig. 5**). This effectively removes all primer dimers, but also ignores any bona fide longer indels. ampliCan instead tries to estimate the likely length of a primer dimer deletion by taking the length of the amplicon, subtracting the primer lengths with a small buffer (30 bp) to arrive at maximally allowed deletion length. This results in a more realistic estimate of the length of artificial deletions that would result from primer-dimers. As an example, for an amplicon of size 150 with primers of length 20, the maximum deletion length would be 80 bp ($150 - (2 \times 20 + 30)$).

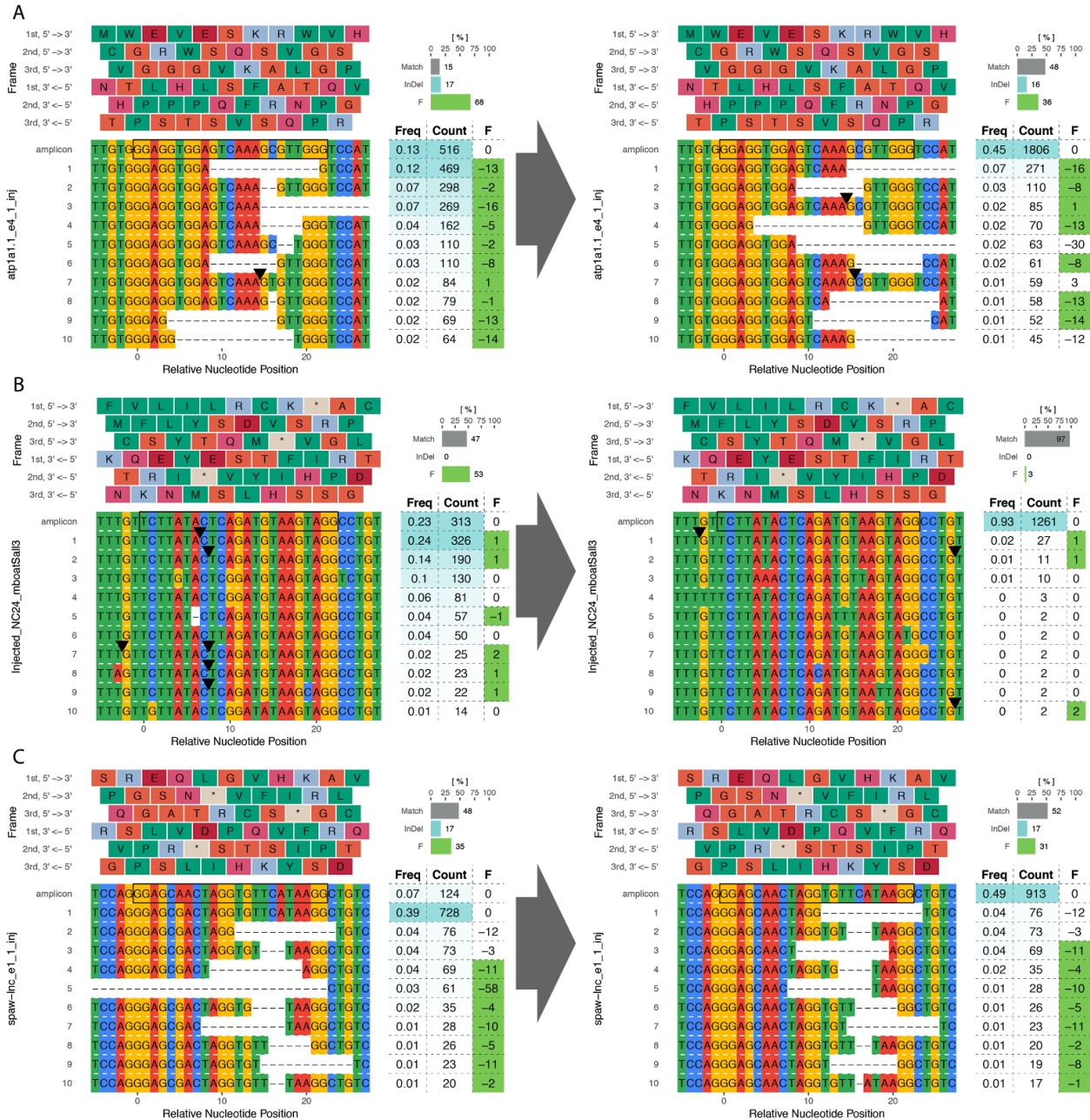
3. Erroneously assigned reads and sequencing artifacts

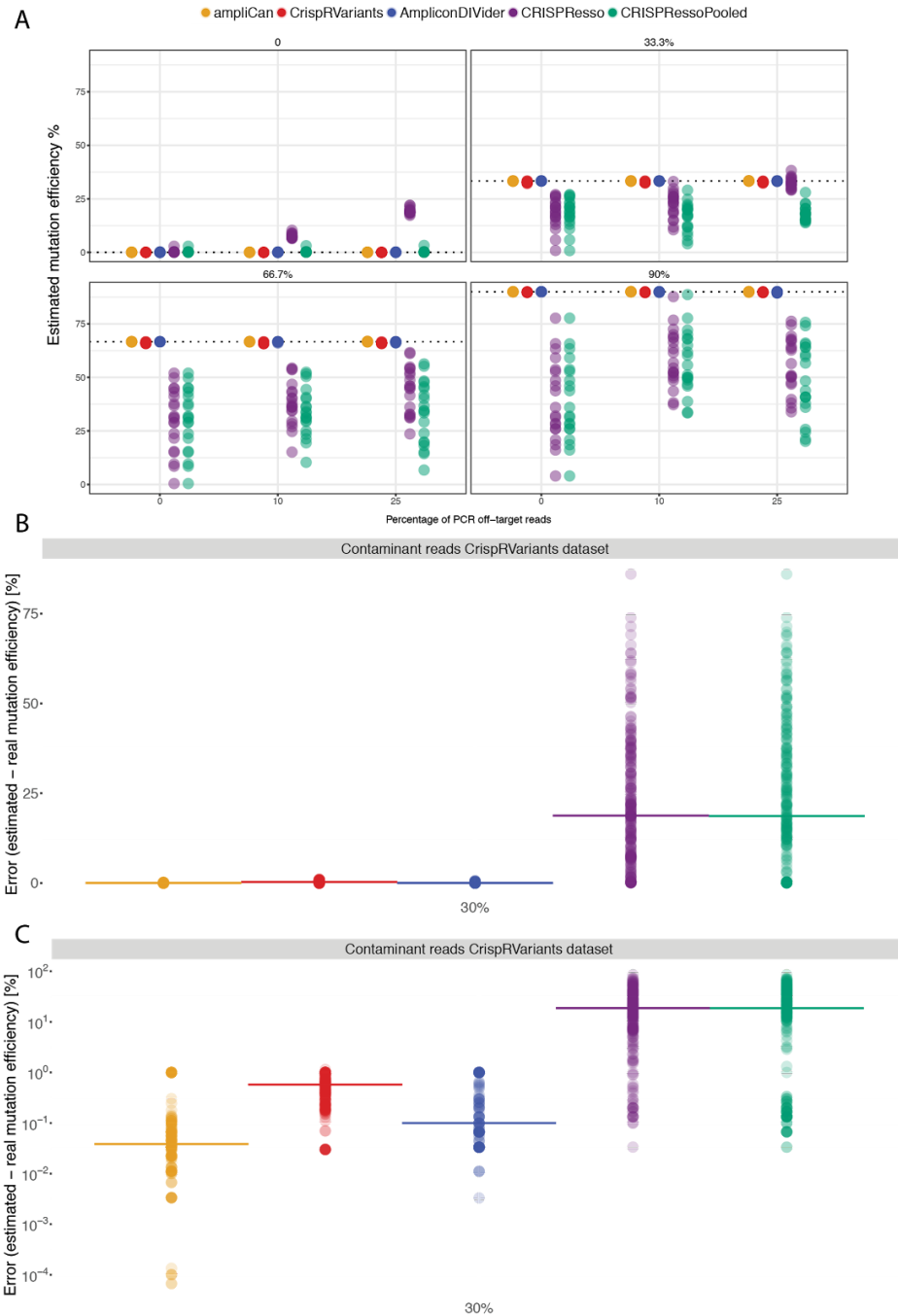
An assumption of ampliCan is that a CRISPR editing event will result in a low number of indel events resulting in a good alignment with few discrepancies to the reference sequence or (if available) the control experiment. To accomplish this ampliCan uses a two dimensional clustering method based on sequence alignment score and sequence alignment indel events to filter out erroneous reads and sequencing artifacts. This takes all alignments and performs k-means clustering with different 1-3 clusters. It then uses the silhouette criterion to determine the optimal number of clusters. In the case of 1 cluster, all reads are either edited or perfectly matching the reference/control. In the case of 2 you have both edited and unedited reads. In the case of 3 clusters, cluster with center that has the biggest number of events and lowest alignment score (in normalized relation on 0-1 scale) means that you in addition have a group of sequencing artifacts or reads that poorly align to the loci (example in **Supplementary Fig. 11**).

ampliCan provides plots that shows the impact of each of the filtering steps, across the whole library for read quality (**Supplementary Fig. 22**) and for each experiments for primer-dimer and assignment/artifacts issues (**Supplementary Fig. 8**).

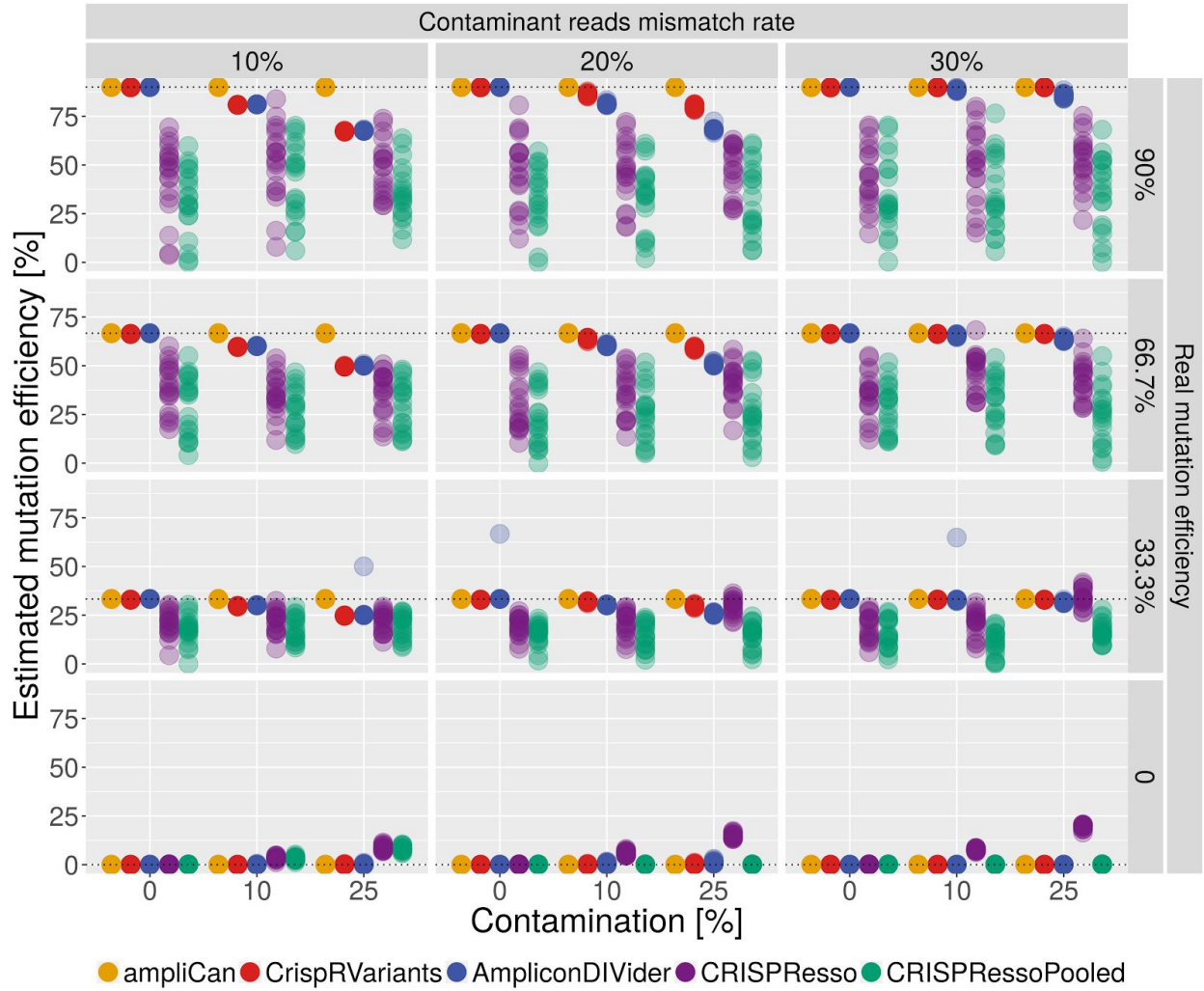
Supplementary Note 7: Read assignment

ampliCan assigns reads to the respective experiment by matching primers used in the amplification of the loci. These region should be immutable and match the reads since an indel spanning a primer would either result in failure to amplify the locus or be “corrected” by the primer when it amplifies the target site. However, since small sequencing and primer synthesis errors could potentially occur in the primer part ampliCan allows for up to 2 mismatches (user customizable) between the primers and reads. During this process it is possible that some reads will be unassigned and not match any of the experiments. While these reads are typically noise from the high-throughput nature of the sequencing experiment, they could in some cases be helpful in troubleshooting failed experiments. ampliCan therefore provides human readable alignments of the top 5 most abundant forward and reverse read pairs aligned to each other (**Supplementary Fig. 15**). In some cases these correspond to off-target PCR amplicons and close homologous regions as well as errors in the specification of the experiment.

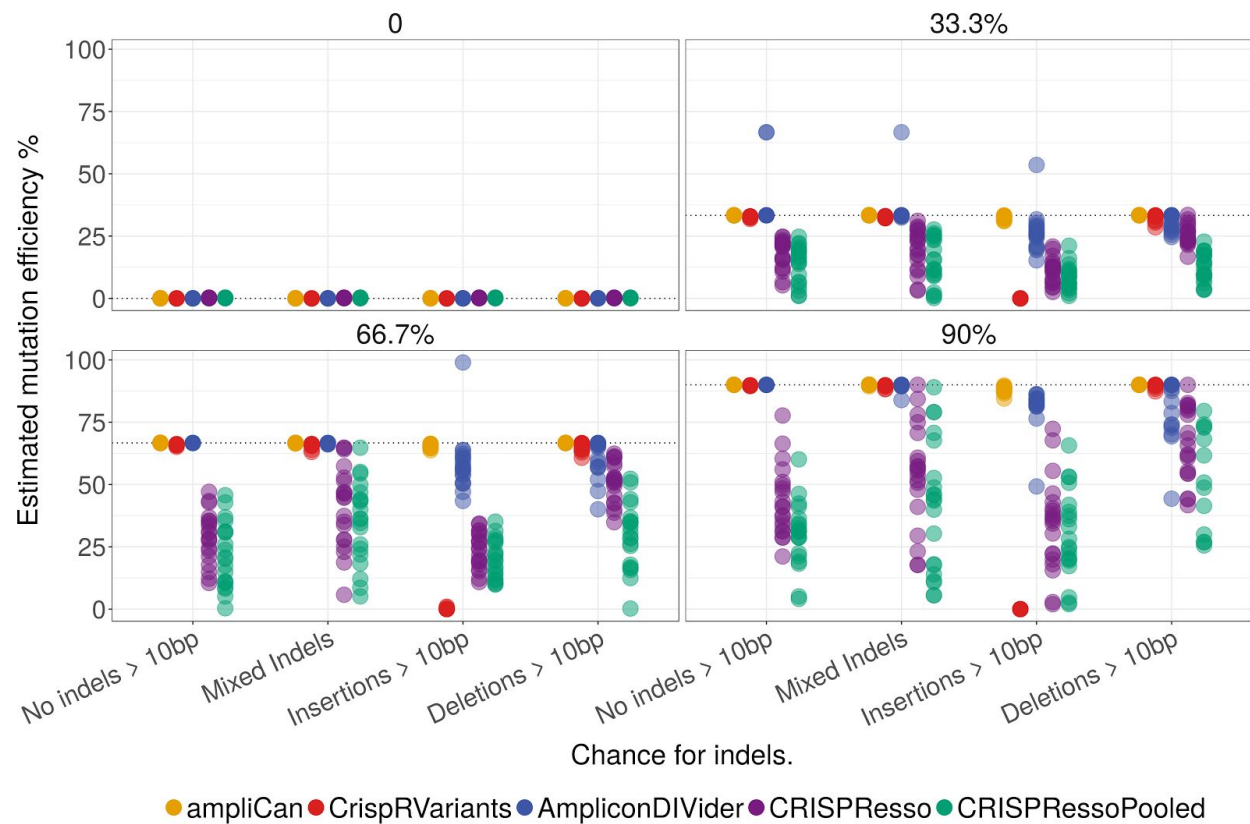




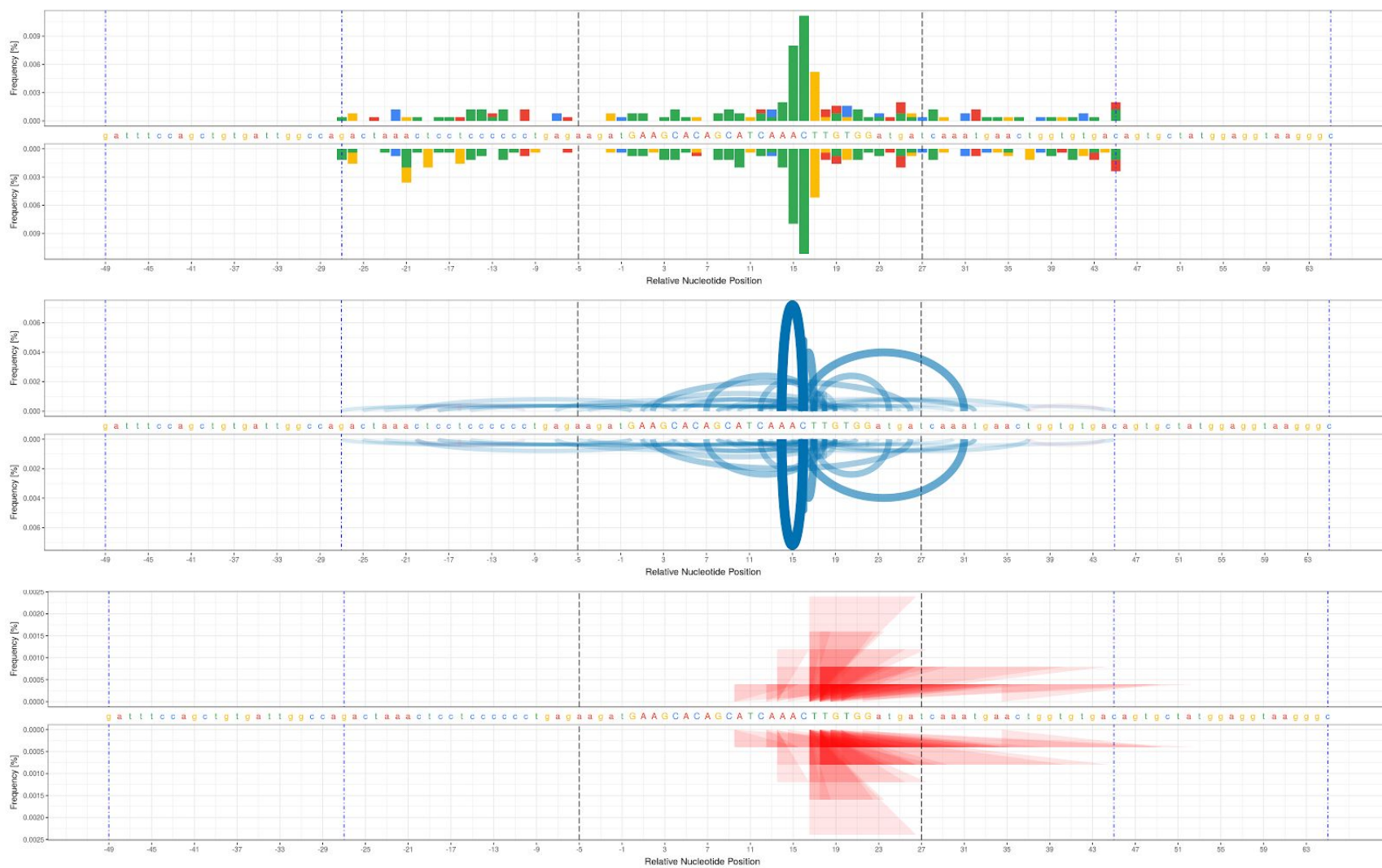
Supplementary Fig 3. A. Performance of leading tools when facing different mutation efficiencies and fractions of contaminating reads. These sets are obtained from the benchmark data in Lindsay et al. (CrispRVariants)¹, Synthetic Dataset 2 (Supplementary Fig. 10 in Lindsay et al.¹). Each dot in the plot correspond to the estimated mutation efficiency calculated by a single tool for a single experiment, while the dotted line shows the true mutation efficiency. The fraction of contaminant reads varies on the x-axis. **B.** Error rates (**C.** log10 scaled) of the same experiments. The median error is indicated by the horizontal line.



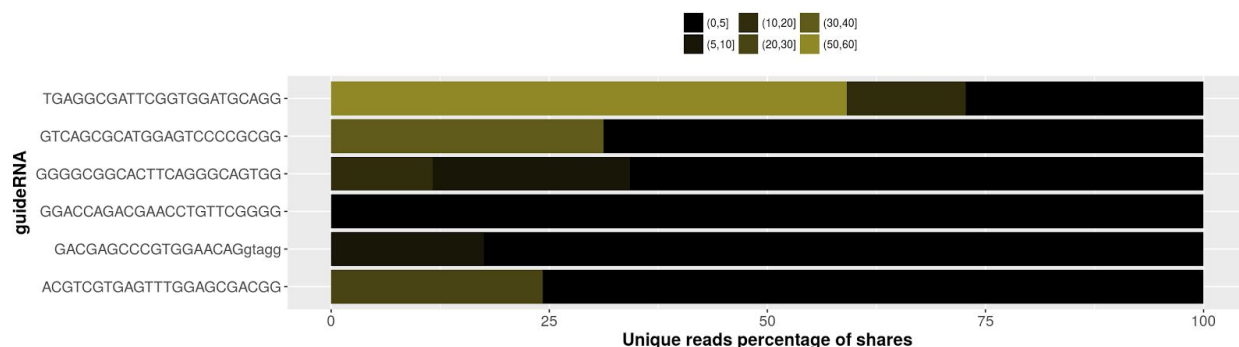
Supplementary Fig 4. The data from left panel of Fig. 2 split by the true mutation efficiency and with a baseline of 0% true mutation efficiency added for comparison. The dots shows estimated mutation efficiency by leading tools on reads with increasing contamination (as 0, 10 and 25 percentage of all reads) with different mismatch rate of the off-target reads (10%, 20%, 30%). Each point corresponds to one experiment. True mutation efficiency is indicated with dotted lines, and labelled to the right of the charts. Contamination is simulated by introducing random mismatches (Contaminant read mismatch rate) in reads mapping to the loci, similar to the benchmark in Lindsay et al.¹. The mismatch rate is indicated at the top. Only ampliCan shows robustness to the whole range of different mismatch rates and mutational efficiencies.



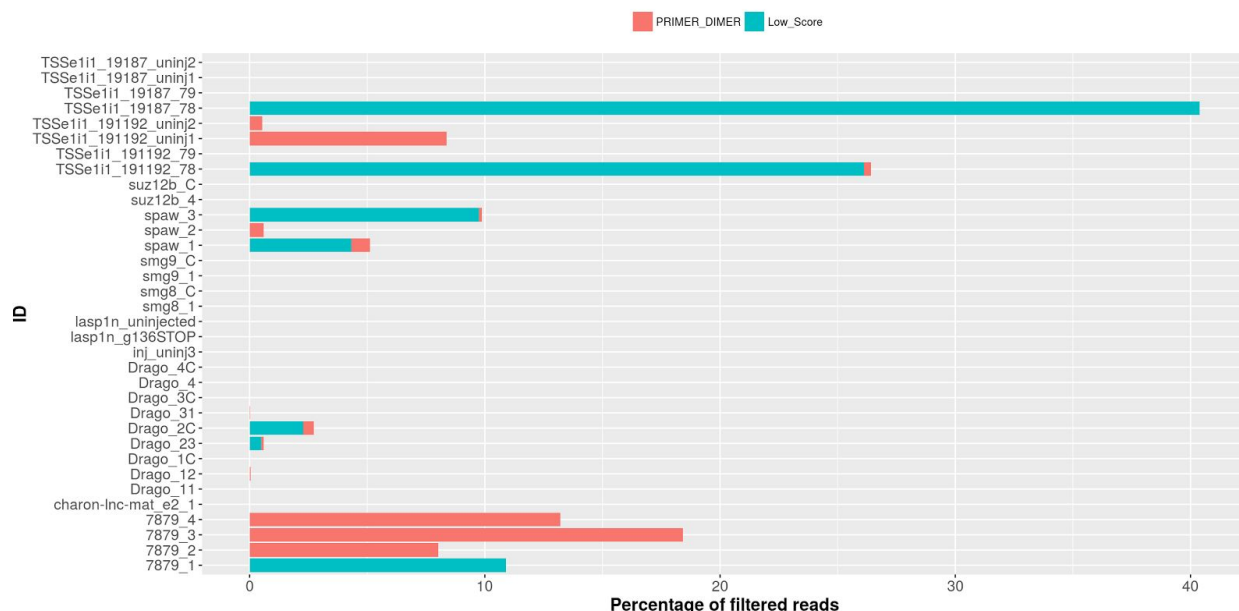
Supplementary Fig 5. Performance of leading tools grouped on simulated data with large indels. Some tools find larger indels challenging. ampliCan consistently returns values closest to the real mutation efficiency.



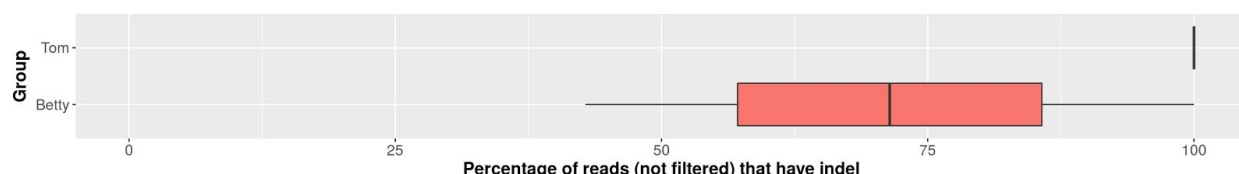
Supplementary Fig 6. A number of plots are produced by ampliCan to investigate editing events: mismatches to the reference (**A**), aggregated deletions (**B**) and aggregated insertions (**C**). Dashed blue vertical lines indicate where the primers start and end. The black dashed lines mark the region around the cut site where we require a mutation event to overlap to be considered a CRISPR event. The top half of the plots shows the forward reads, while the bottom shows the reverse reads. The amplicon sequence is shown in the middle, with uppercase letters indicating the gRNA. The y axis shows the efficiency of the events summed over all reads. This allows for immediately discerning which bases are edited above the background noise from sequencing.



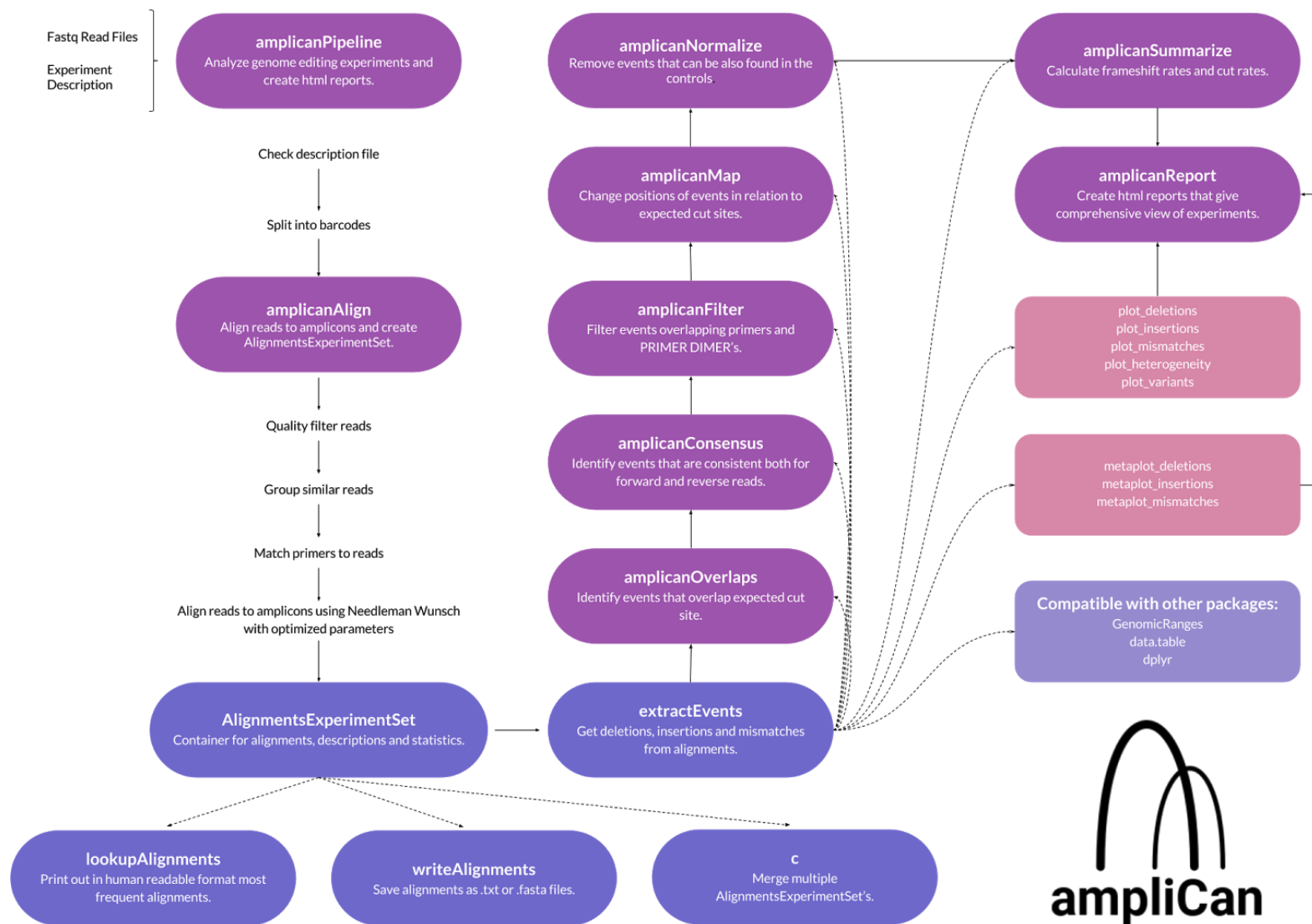
Supplementary Fig 7. Example of a heterogeneity plot produced by ampliCan. In this, identical reads are collapsed together and grouped by gRNA. A stronger shade of yellow indicates a large group of homogeneous reads. This plot can give insight into the heterogeneity of the outcome. A high level of heterogeneity can indicate sequencing problems or mosaicism. Reads can also be aggregated on experiments or any other user selected grouping rather than guides.



Supplementary Fig 8. Example of bar plot showing fraction of reads that were filtered out of the experiments. Red bars correspond to a primer-dimer filter, and blue bars indicate low quality reads.

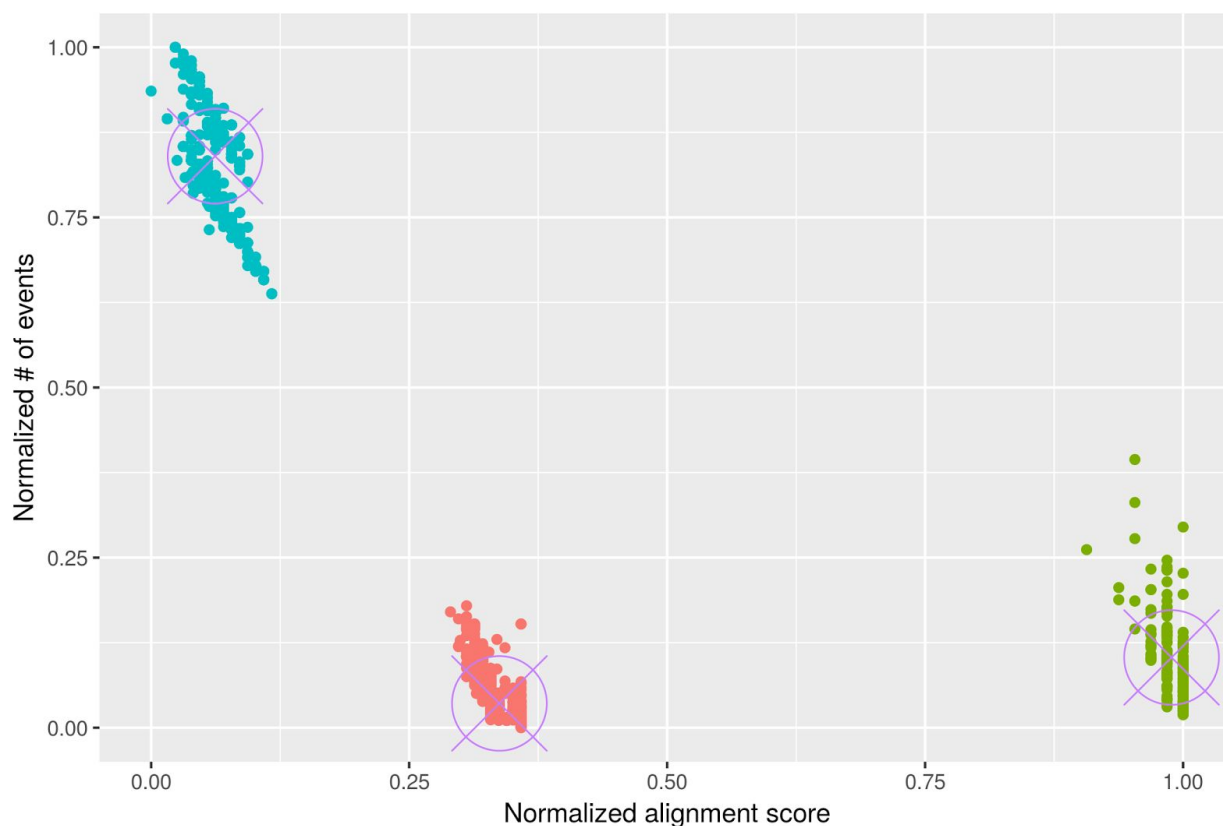


Supplementary Fig 9. Example of comparison plots in ampliCan test data where Indel rates are grouped by the researcher performing the experiments. These can be grouped on any user specified criteria.

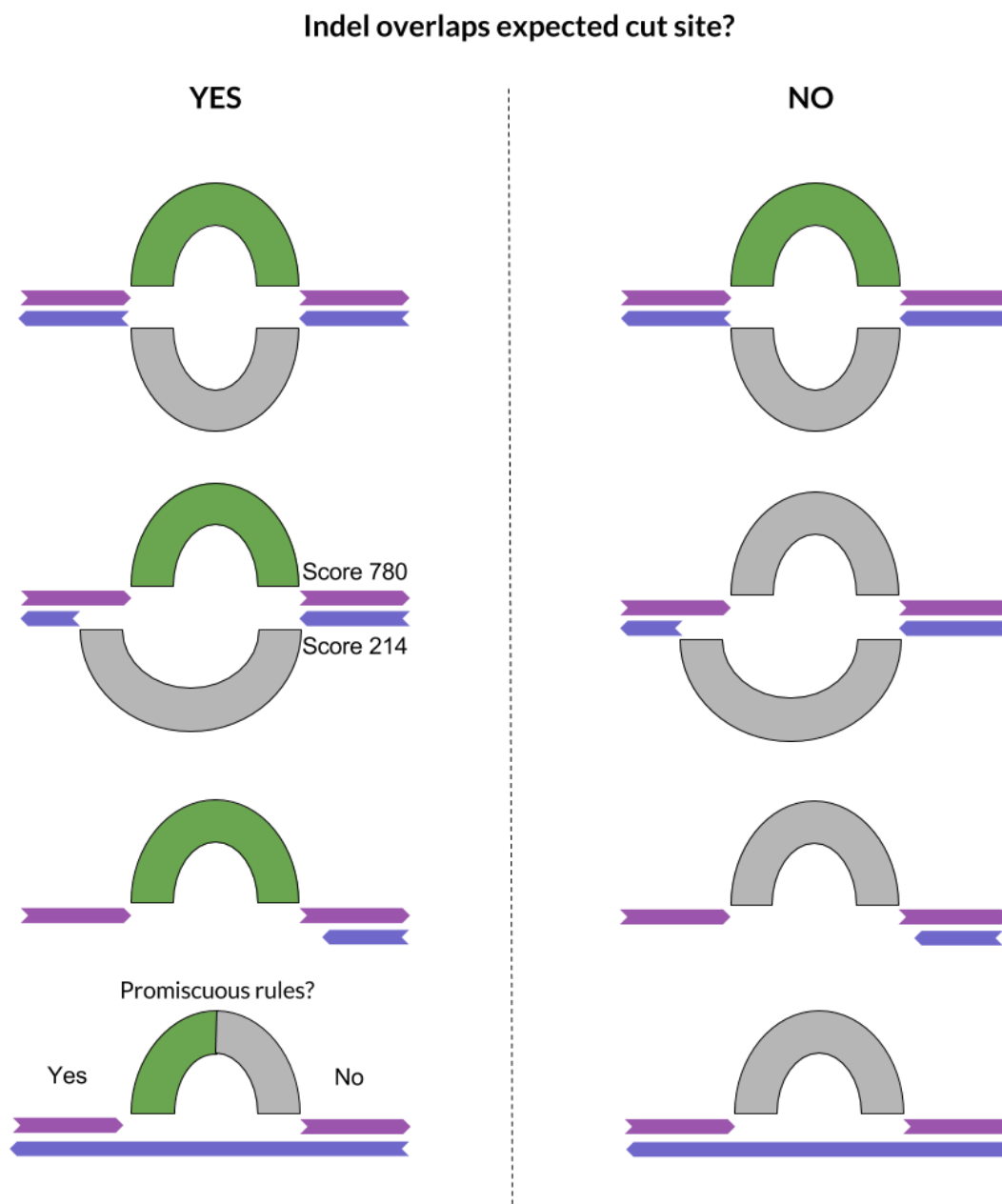


Supplementary Fig 10.

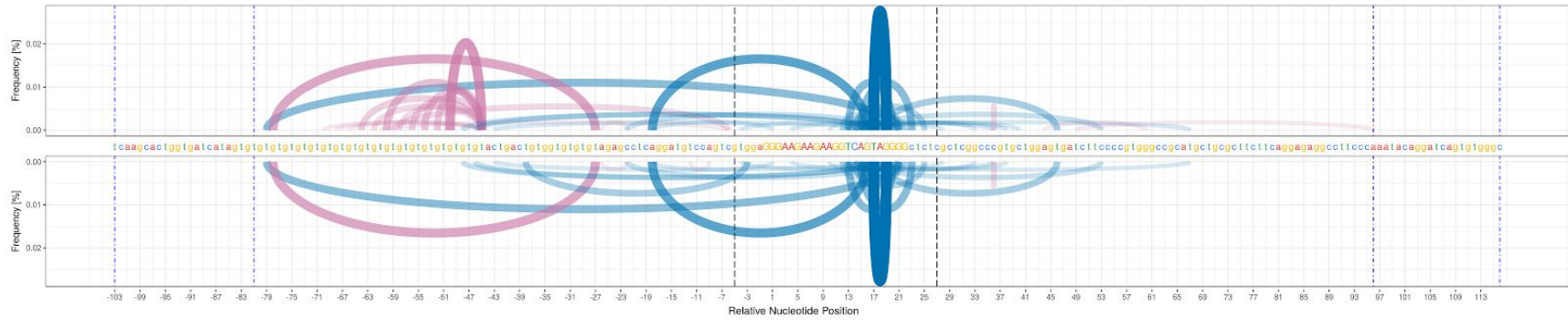
Overview of the ampliCan pipeline



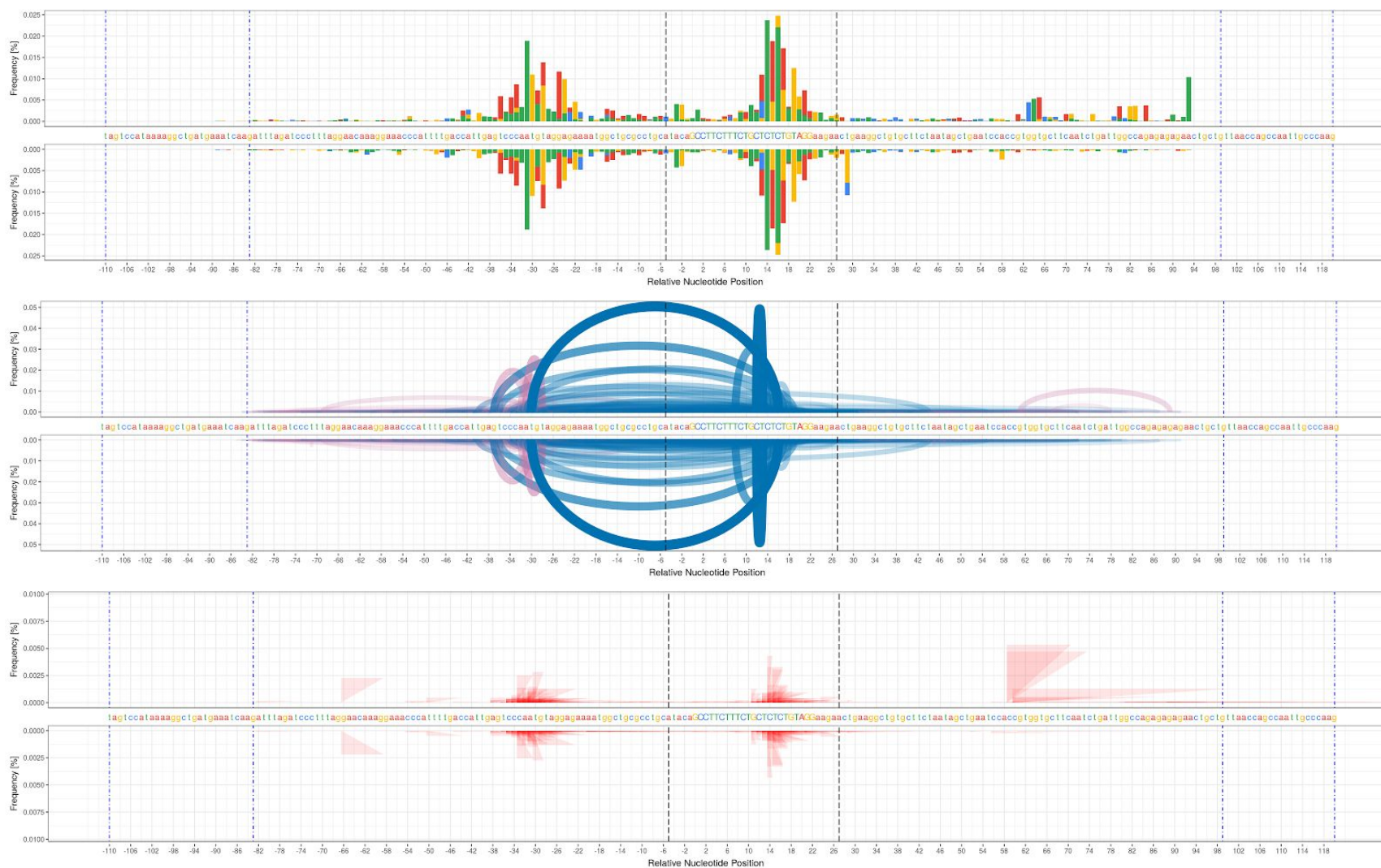
Supplementary Fig 11. Example of k-means clustering of reads during filtering of contaminant reads. A low alignment score (x axis) combined with a high number of events (y axis) indicate erroneous reads. Silhouette criterion is used to determine whether data should be clustered into two (read with no edits and reads with editing events) or three clusters (a noise cluster). In the case of three clusters, the cluster with its center (purple) closest to the upper left corner is filtered.



Supplementary Fig 12. The paired-end read consensus rules for ampliCan. Events marked in green are considered real cut sites while those in gray are not. When reads from forward (purple) and reverse (blue) reads are in agreement there is a consensus (top row). In situations where an event is only covered by one read, that read is preferred. When events from the two reads are in disagreement, but overlap each other, the event from the strand with a higher alignment score will be used. In cases where one of the strands ends prematurely we treat the situation as if it only has one strand. In rare case where there are events on one strand and the other has continuous alignment ampliCan allows users to define the behaviour (default is promiscuous rule enabled).



Supplementary Fig 13. ampliCan evaluates forward and reverse reads separately which allows users to spot potential problems. Here, purple colour indicates cuts outside of the gRNA expected cut site, a large majority of those reads comes from problematic alignments, where forward read and reverse read are in disagreement.



Supplementary Fig 14. Mismatch plot (**top**), deletion plot (**middle**) and insertion plot (**bottom**). This presents a case where the user might have mischaracterized their own experiment which can be spotted with these plots. Here nickases were used, but only one of the gRNAs was specified in the configuration file. A bimodal distribution of mismatch events and

insertions is the result. A quick examination allows the user to realize the expected cut site should be extended to include the second gRNA, enabling more precise mutation efficiency estimation.

1 Description
2 Barcode Summary
3 Top unassigned reads
3.1 barcode_1

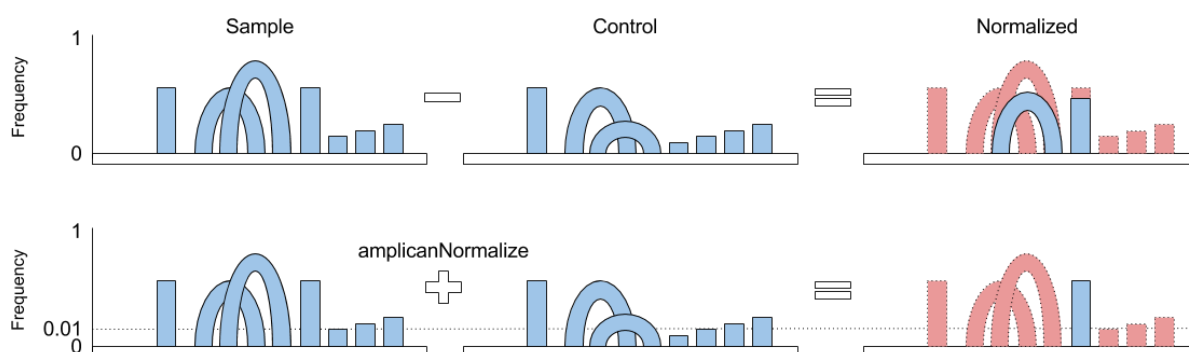
3 Top unassigned reads

3.1 barcode_1

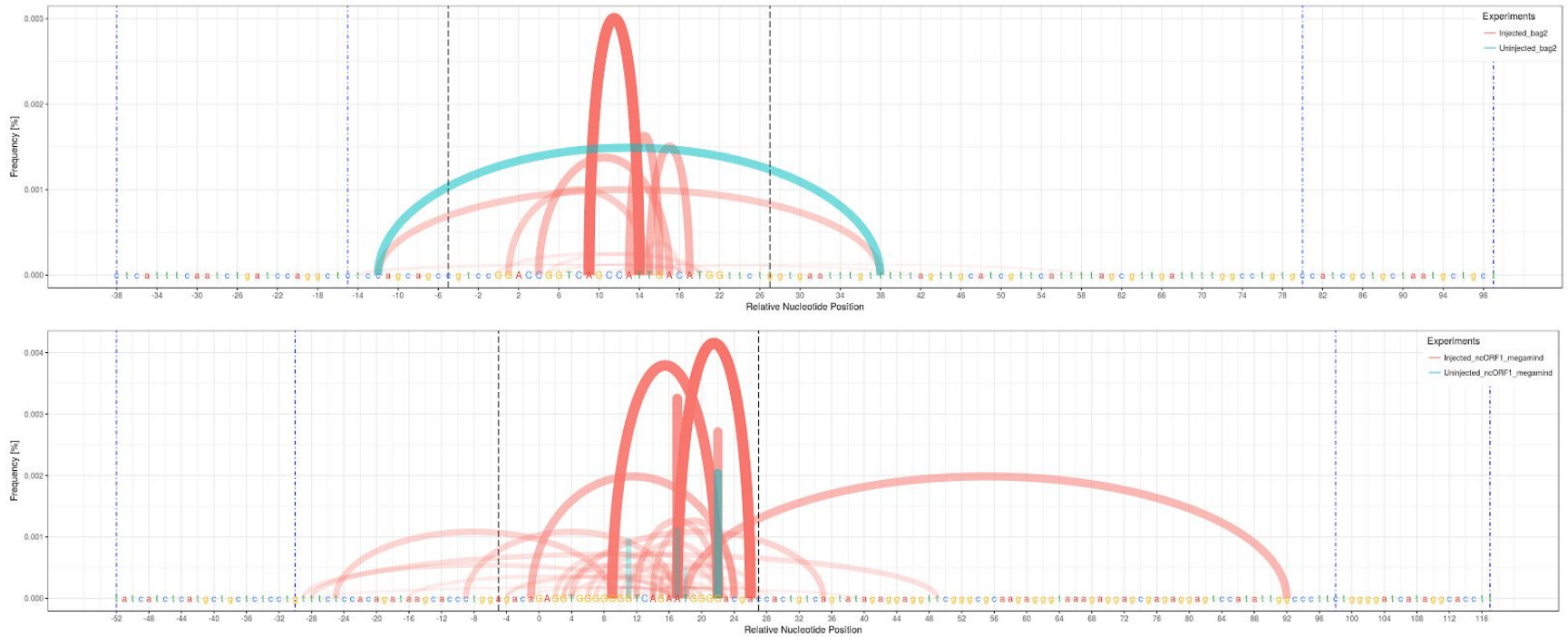
Forward	Reverse	Counts	Frequency
P1	S1	1	0.0714286

P1	1	AAAT--ACTGTCTGTGACCAAACTTCTTAAGGTGCTGTTTT-GATGAT	47
S1	1	AAGCTGACGGCTAAATGA--AAAATATCTGAAACATCTGTTCCAGGTGCT	48
P1	48	AAACTTTATTGTGCTTTTGTAGTTGTGCCCTTGTGTTGGCAGAGGGTCA	97
S1	49	GCGTATGCCAGGGCAGA-GAAGAAG-GTCAGGGAAGTCACTGGAGGTCA	96
P1	98	--GCAGACCAGTAAGTCTTCTCAATTTCTTTTATTATGTATATGTAGT	144
S1	97	CTGGGATACCCTT---TCTTCCACACCAATGGGGAAGGAGTCTTGCCA	143
P1	145	GATAAA-A	151
S1	144	GATGACCA	151

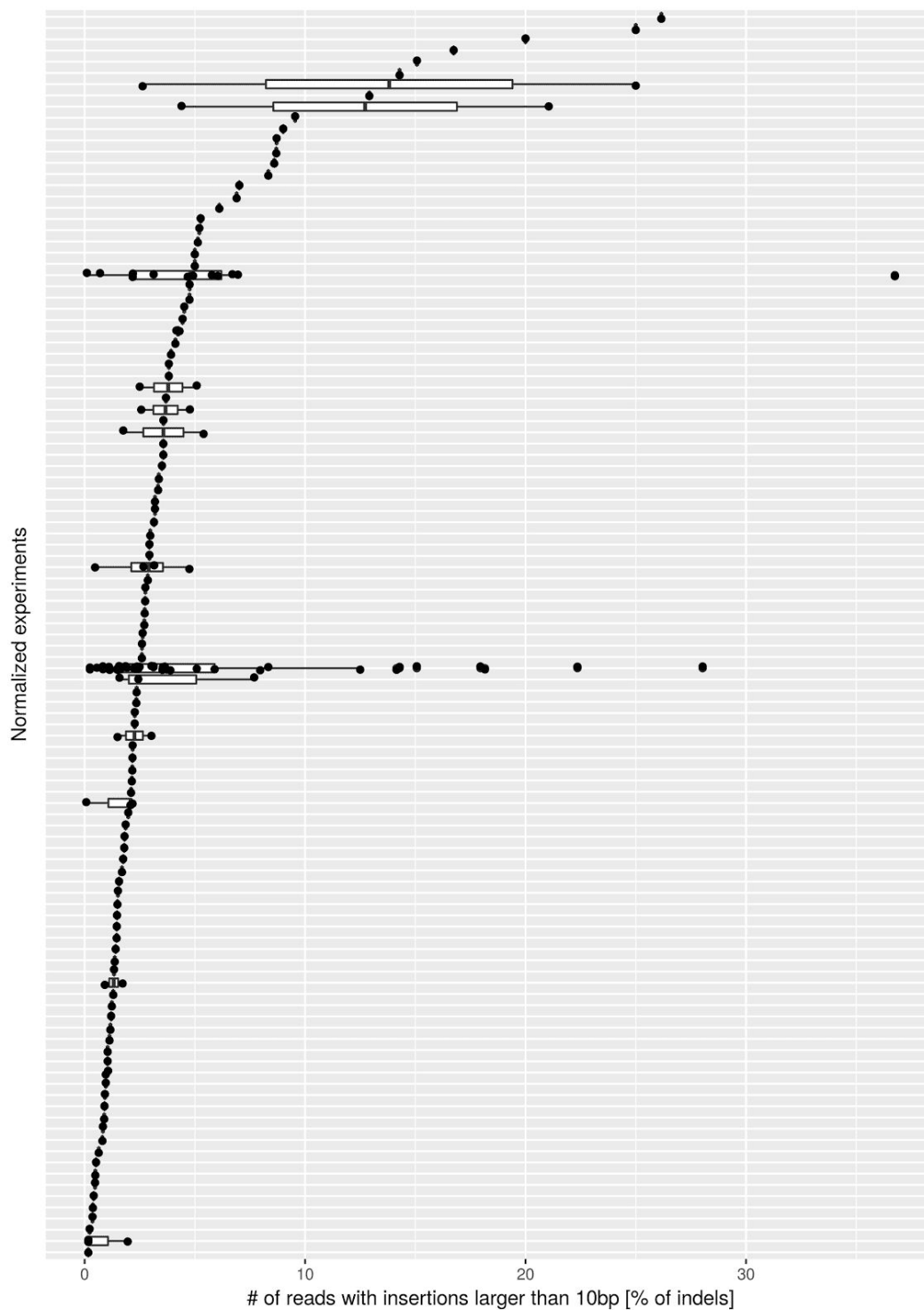
Supplementary Fig 15. Screenshot of the example barcode report, top unassigned read section. Human readable alignment of forward and reverse reads of the top most frequent unassigned reads is presented. Huge fragmentation and poor alignment suggest contamination, while low frequency indicate proper specification of the primers in the config file.



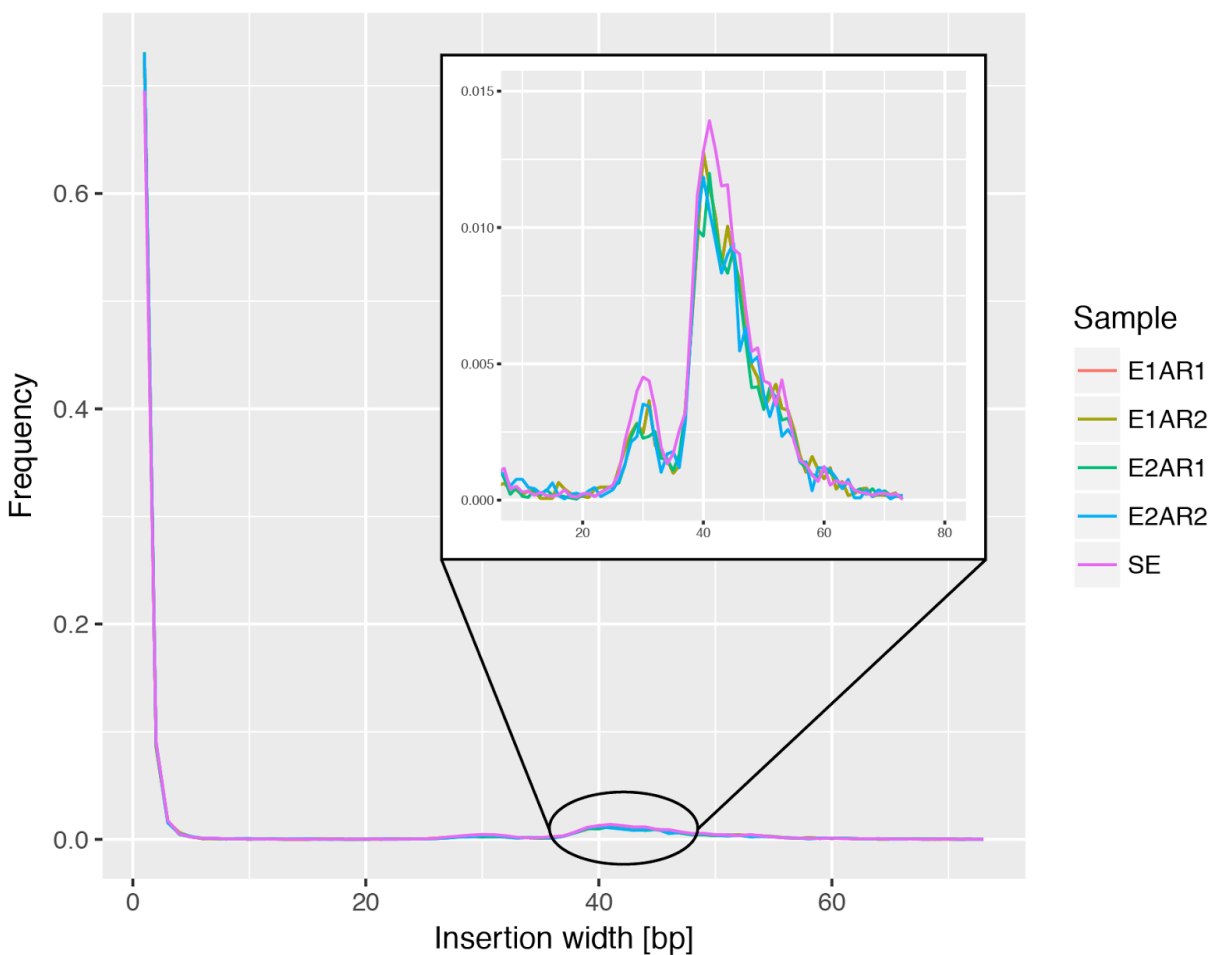
Supplementary Fig 16. Two common methods for normalization using controls. In the first, the frequencies from events in the control sample are subtracted from the frequencies in the treated sample (subtraction method). In the second method, all events occurring in the control above a frequency threshold (ampliCan default: 0.01) are removed from the treated sample. The latter is more robust to stochastic differences.



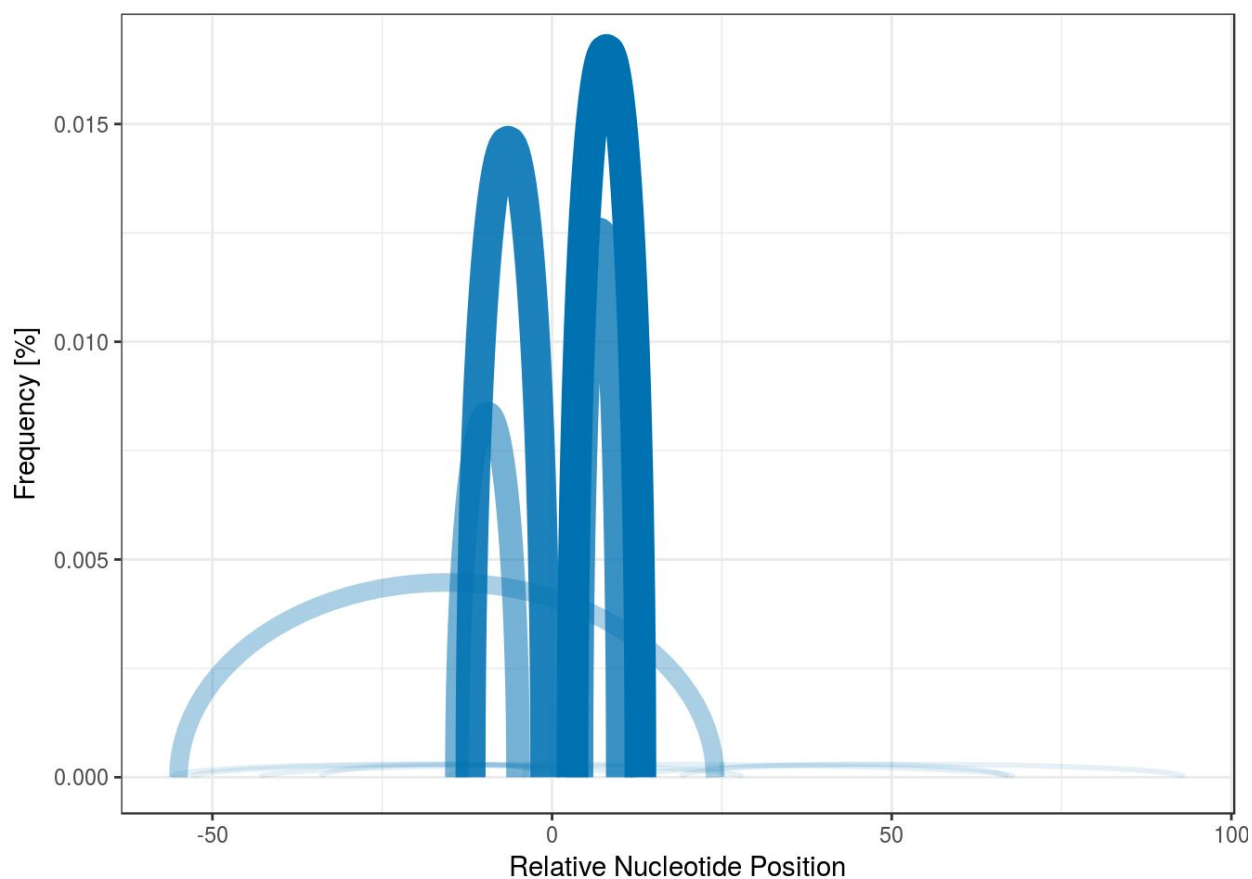
Supplementary Fig 17. Deletion plots produced by ampliCan for two experiments showing deletions in Cas9 injected (red) versus control (cyan) samples. The archs indicate deletions (x-axis, start to end of arch) present in the samples at a frequency indicated by the y-axis and transparency. The blue, vertical dotted lines shows the start and end of the primers. In the top panel experiment normalization using the subtraction method would filter the big deletion also present in the control, but it would not completely get rid of the 1bp long deletion in the lower panel experiment. Both cases would get expunged with the default ampliCan normalization.



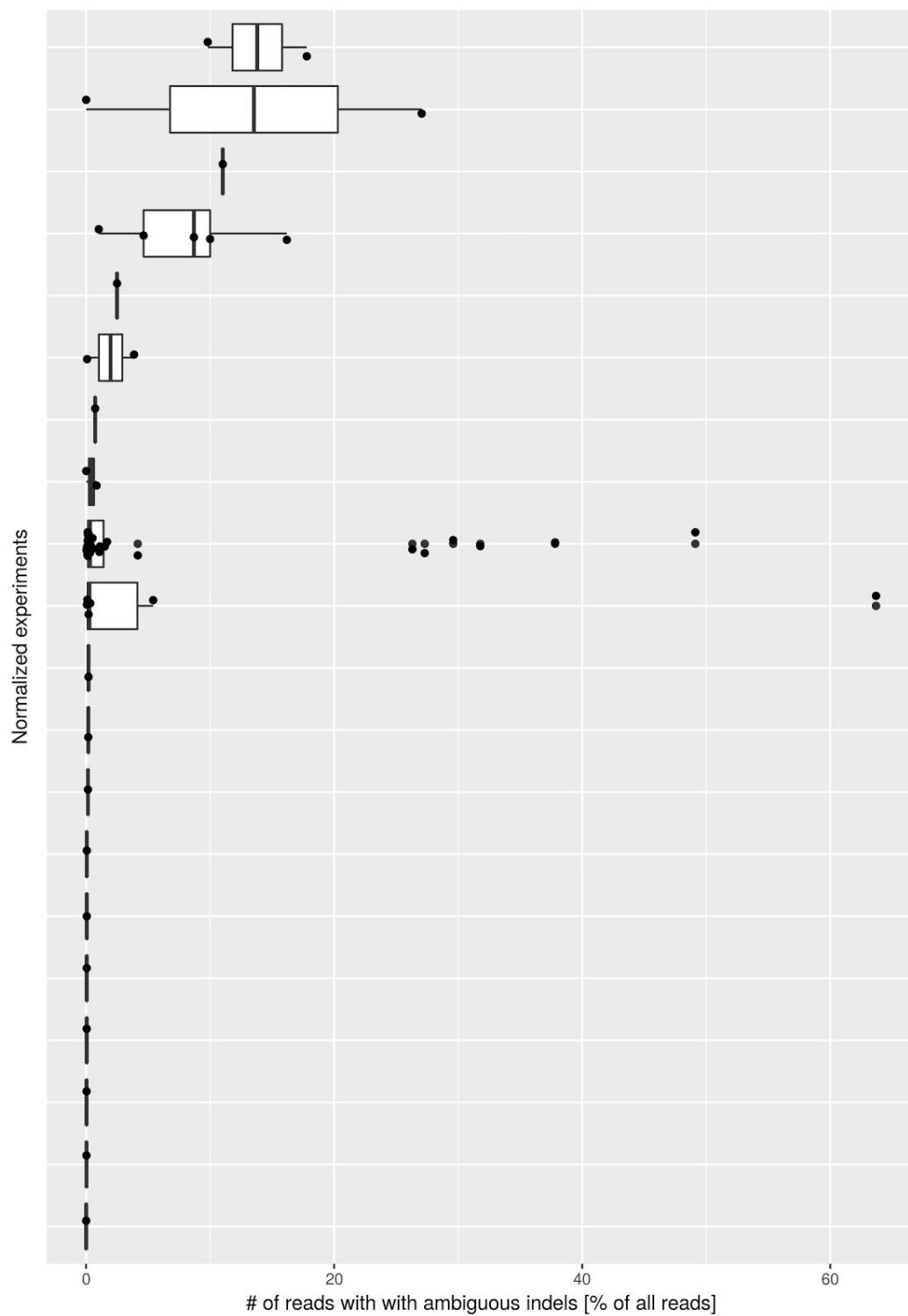
Supplementary Fig 18. Fraction of reads having indels greater than 10 bp across 176 experiments. Each dot represents one experiment and are grouped in rows by having the same gRNA (replicates). All experiments are normalized using wild type controls ensuring that these are real indel events. The higher mean for some of the replicated experiments suggests that some gRNAs have a higher chance of resulting in long indels.



Supplementary Fig 19. Distribution of insertions width for lentiviral samples on Chari et al. datasets. Shows the proportion (y-axis) of reads with a given insert length (x-axis). The insert shows a population of unintended larger insertions. Around 90% of these originate from the lentiviral vector used to transfect the cells.

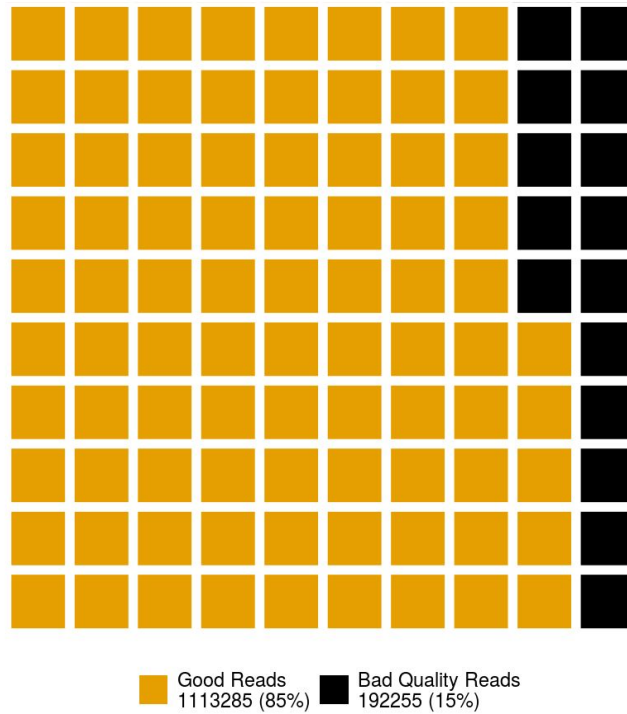


Supplementary Fig 20. Example of a deletion metaplot produced by ampliCan summarizing multiple experiments. Here, an aggregation of editing events from many experiments (many targets) using the same gRNA is presented giving an overall gRNA cut profile. Position 0 is relative to the first 5' base of gRNA.



Supplementary Fig 21. Percentage of reads with ambiguous indels caused by disagreement of forward and reverse reads. ampliCan consensus rules help to mitigate mis-estimation that could arise from these events.

Quality of all reads



Supplementary Fig 22. Waffle plot of the quality of reads across all experiments.

Supplementary Tables

Tool	Aligner	Notes
ampliCan	Needleman-Wunsch from Biostrings ⁶ as local alignment	
CrispRVariant ¹	BWA-MEM ⁷ as global alignment	Filters primer dimers by restricting start/end of indel, but may result in missing larger deletions. Larger insertions can be missed through mapping.
ampliconDIVider ⁸ (ampliconDIV_minimal.sh)	BWA-MEM as global alignment	Sometimes returns estimates above 100%, these values were filtered. We run only the variant counting step of the pipeline. The full pipeline requires the commercial novoAlign.
CRISPResso & CRISPRessoPooled ⁹	BWA-MEM + FLASH (in Pooled) as local alignment	Poor performance due to unknown issues. It was installed and run in the same fashion as in Lindsay et al. analysis. CRISPRessoPooled uses local-global alignment strategy.

Supplementary Tab 1. Table of compared tools, with align strategy.

```
ID: ID_1 read_id: 1 Count: 3
AAGCTGACGGCTAAATGAAAAATGTCAACGCTCTGTTCCAG-----AAAAAAAAAAAAAAAAAATCCACACCAATGGGAAAGGAGTCTGCCAGATGACCATCC
CAACTGTGTTGCAGCAGCCAGATCCAGGTGTGTTGCGCTTGTGTAATT
AAGCTGACGGCTAAATGAAAAATGTCAACATCTGTTCCAGGTGCTGCGTATGCCAGGGCAGAGGAGGTGTCAGGGAACCTGGAGGTCACTGGGATACCCCTTC-----TTCCACACCAATGGGAAAGGAGTCTGCCAGATGACCATCC
CAACTGTGTTGCTGCAGCCAGATCCAGGTGTGTTGCGCTTGTGTAATT

AAGCTGACGGCTAAATGAAAAATGTCAACGCTCTGTTCCAG-----AAAAAAAAAAAAAAAAAATCCACACCAATGGGAAAGGAGTCTGCCAGATGACCATCC
CAACTGTGTTGCAGCAGCCAGATCCAGGTGTGTTGCGCTTGTGTAATT
AAGCTGACGGCTAAATGAAAAATGTCAACATCTGTTCCAGGTGCTGCGTATGCCAGGGCAGAGGAGGTGTCAGGGAACCTGGAGGTCACTGGGATACCCCTTC-----TTCCACACCAATGGGAAAGGAGTCTGCCAGATGACCATCC
CAACTGTGTTGCTGCAGCCAGATCCAGGTGTGTTGCGCTTGTGTAATT

ID: ID_1 read_id: 2 Count: 2
AAGCTGACGGCTAAATGAAAAATGTCAACGCTCTGTT-----AAAAAAAAAAAAAAAAAATCCACACCAATGGGAAAGGAGTCTGCCAGATGACCATCC
CAACTGTGTTGCAGCAGCCAGATCCAGGTGTGTTGCGCTTGTGTAATT
AAGCTGACGGCTAAATGAAAAATGTCAACATCTGTTCCAGGTGCTGCGTATGCCAGGGCAGAGGAGGTGTCAGGGAACCTGGAGGTCACTGGGATACCCCTTC-----CACACCAATGGGAAAGGAGTCTGCCAGATGACCATCC
CAACTGTGTTGCTGCAGCCAGATCCAGGTGTGTTGCGCTTGTGTAATT

AAGCTGACGGCTAAATGAAAAATGTCAACGCTCTGTT-----AAAAAAAAAAAAAAAAAATCCACACCAATGGGAAAGGAGTCTGCCAGATGACCATCC
CAACTGTGTTGCAGCAGCCAGATCCAGGTGTGTTGCGCTTGTGTAATT
AAGCTGACGGCTAAATGAAAAATGTCAACATCTGTTCCAGGTGCTGCGTATGCCAGGGCAGAGGAGGTGTCAGGGAACCTGGAGGTCACTGGGATACCCCTTC-----CACACCAATGGGAAAGGAGTCTGCCAGATGACCATCC
CAACTGTGTTGCTGCAGCCAGATCCAGGTGTGTTGCGCTTGTGTAATT
```

Supplementary Tab 2. Example of human readable output. Aligned reads are assigned to the experiment (ID, read_id) and sorted based on count (Count). For each pair alignment is presented with top part representing forward read aligned to amplicon and bottom presenting reverse read aligned to amplicon.

seqnames	start	end	width	strand	originally	replacement	type	read_id	score	counts
ID_1	108	127	20	+		AAAAAAAAAAAAAAAAAAAAA	insertion	1	597	3
ID_1	112	131	20	+		AAAAAAAAAAAAAAAAAAAAA	insertion	2	557	2
ID_1	42	107	66	+			deletion	1	597	3
ID_1	38	111	74	+			deletion	2	557	2
ID_1	24	173	150	+			deletion	3	193	1
ID_1	34	117	84	+			deletion	4	532	1
ID_1	31	31	1	+	A	G	mismatch	1	597	3
ID_1	163	163	1	+	T	A	mismatch	1	597	3
ID_1	31	31	1	+	A	G	mismatch	2	557	2
ID_1	163	163	1	+	T	A	mismatch	2	557	2
ID_1	23	23	1	+	T	G	mismatch	3	193	1
ID_1	176	176	1	+	A	T	mismatch	3	193	1
ID_1	177	177	1	+	G	C	mismatch	3	193	1
ID_1	31	31	1	+	A	G	mismatch	4	532	1
ID_1	171	171	1	+	G	A	mismatch	4	532	1
ID_1	108	127	20	-		AAAAAAAAAAAAAAAAAAAAA	insertion	1	597	3
ID_1	112	131	20	-		AAAAAAAAAAAAAAAAAAAAA	insertion	2	557	2
ID_1	42	107	66	-			deletion	1	597	3
ID_1	38	111	74	-			deletion	2	557	2
ID_1	24	173	150	-			deletion	3	193	1
ID_1	34	117	84	-			deletion	4	532	1
ID_1	31	31	1	-	A	G	mismatch	1	597	3
ID_1	163	163	1	-	T	A	mismatch	1	597	3
ID_1	31	31	1	-	A	G	mismatch	2	557	2
ID_1	163	163	1	-	T	A	mismatch	2	557	2
ID_1	23	23	1	-	T	G	mismatch	3	193	1
ID_1	176	176	1	-	A	T	mismatch	3	193	1
ID_1	177	177	1	-	G	C	mismatch	3	193	1
ID_1	31	31	1	-	A	G	mismatch	4	532	1
ID_1	171	171	1	-	G	A	mismatch	4	532	1
ID_2	115	127	13	-			deletion	1	845	3
ID_2	106	114	9	-			deletion	2	865	2
ID_2	101	105	5	-			deletion	3	885	1
ID_2	115	127	13	+			deletion	1	845	3
ID_2	106	114	9	+			deletion	2	865	2
ID_2	101	105	5	+			deletion	3	885	1
ID_3	171	171	1	+		T	insertion	1	819	5
ID_3	74	89	16	+		CCCCCCCCCCCCCCCCC	insertion	3	860	1
ID_3	66	83	18	+			deletion	1	819	5

Supplementary Tab 3. Example GenomicRanges table output with additional meta-columns. This representation of alignments allows for efficient manipulation and processing of the data.

ID	Barcode	Forward_Reads	Reverse_Reads	Group	Control	guideRNA	Forward_Primer	Reverse_Primer	Direction	Amplicon
ID_1	barcode_1	R1_001.fastq	R2_001.fastq	Betty	0	AGGTGGTCAGGGAAGTGG	AAGCTGACGGCTAAATGA	AATTACACAAGCGCAACACAC	0	...cagaggAGGTGGTCAGGGAAGTGGga...
ID_2	barcode_1	R1_001.fastq	R2_001.fastq	Tom	0	TGACCCTCTGCCAACACAAGGGG	TGACCAACCTTCTTAAGGTGC	CTCTGCTGCAAAATGCAAGG	1	...agtgtgCCCTTGTGTTGGCAGAGGGTCag...
ID_3	barcode_2	R1_002.fastq	R2_002.fastq	Tom	0	AGGTGGTCAGGGAAGTGG	AAGCTGACGGCTAAATGA	AATTACACAAGCGCAACACAC	0	...ggAGGTGGTCAGGGAAGTGGga...
ID_4	barcode_2	R1_002.fastq	R2_002.fastq	Betty	0	GTCCCTGCAACATTAAAGGCCGG	GCTGGCAACATTCTACCAGT	GAGCGCTGAGGCAGGATTAT	0	...ccaGTCCCTGCAACATTAAAGGCCGgaag...

Supplementary Tab 4. Example ampliCan config file. ampliCan requires this file as minimal input, together with relevant fastq files. More precise, up to date description of the file can be found in the ampliCan vignettes.

References

1. Lindsay, H. *et al.* CrispRVariants charts the mutation spectrum of genome engineering experiments. *Nat. Biotechnol.* **34**, 701–702 (2016).
2. Chari, R., Mali, P., Moosburner, M. & Church, G. M. Unraveling CRISPR-Cas9 genome engineering parameters via a library-on-library approach. *Nat. Methods* **12**, 823–826 (2015).
3. Yin, T., Cook, D. & Lawrence, M. ggbio: an R package for extending the grammar of graphics for genomic data. *Genome Biol.* **13**, R77 (2012).
4. Wickham, H. *ggplot2: Elegant Graphics for Data Analysis*. (Springer, 2016).
5. Morgan, M. *et al.* ShortRead: a bioconductor package for input, quality assessment and exploration of high-throughput sequence data. *Bioinformatics* **25**, 2607–2608 (2009).
6. Pages, H., Gentleman, R., Aboyoun, P. & DebRoy, S. Biostrings: String objects representing biological sequences, and matching algorithms, 2008. *R package version 2*, 160
7. Li, H. Aligning sequence reads, clone sequences and assembly contigs with

BWA-MEM. *arXiv [q-bio.GN]* (2013).

8. Varshney, G. K. *et al.* High-throughput gene targeting and phenotyping in zebrafish using CRISPR/Cas9. *Genome Res.* **25**, 1030–1042 (2015).
9. Pinello, L. *et al.* Analyzing CRISPR genome-editing experiments with CRISPResso. *Nat. Biotechnol.* **34**, 695–697 (2016).