Supplemental Methods

Validation of *Notch* deletions

For validation of *Notch* deletions, we used the following primers:

Sample P9: TGCACCGACGCGGTCACACTGCCGATTT and

AGGAGATGTCGGCGGAGAAGTAGGAAGAGACG

Sample P17: GCCAAAAGCGATTCCACG and GAATCGCTCTCGTTGTTGGC

Sample P65: CAACTGTTTGCTGTTCTGAAATCC and GGCAGCTATAGTCACCGATCC

Sample D1: TTTAGATTAAGTGCTGTGCTGCATCTGTGAC and

ACTCGAGCTCAGGAAATGCC

Sample D7: AGAGGCCCTTGGGCTATTGGCCTGCTGA and

GCCATCGATGCAGGTTCCTCCGTTCTGGCA

Sample P29: GCCTGCTTTCATCGTTGATTGT and CAAACCGATTTGCCAGGAATGT

Sample P41: CCACCATTCTCCTTGCCAATTC and CCTCTACTCGCTTCTCGGTTTC

Sample P61: CTGCCACGCCTCTTGTTGAC and TGCAGTCAAAAACTGATGAATGAT

PCR reactions were performed using tumour DNA as well as control DNA isolated from the adjacent gut, the head and the thorax of the same fly, using 0.5 – 1 ng of genomic DNA. Thorax controls were not available for samples P9, P17 and P65. For amplicons above 2 kb we used the LongAmp Taq DNA Polymerase (New England Biolabs) in a standard PCR reaction mix. Amplicons below 2 kb were amplified with the Phusion High Fidelity DNA polymerase (Thermo Scientific) using standard conditions. For Sanger sequencing, we gel-purified DNA bands of interest and sequencing was performed by Eurofins Genomics with the same primers as used for the PCR amplification.

Read tagging

We have recently developed a bioinformatic tool named readtagger to tag reads that map to multiple genomes (Siudeja et al. 2021). In cases where reads map to a primary genome with

less affinity than to a secondary genome, reads are tagged as originating from the secondary source, and can then be filtered or extracted from the alignment to the primary genome. Using this approach, we first mapped reads to *Drosophila* reference TE sequences using BWA-MEM v0.7.15 to annotate non-reference TE sequences in the genomes of our fly stocks. This step enables the easy extraction and annotation of clusters of transposable element-mapped reads in downstream analysis. Next, in order to filter out reads that likely originated from contaminating species of the *Drosophila* microbiome, we mapped reads to several known species found in the *Drosophila* gut (*Acetobacter pasteurianus* (NC_013209.1), *Escherichia coli* (NC_000913.3), *Innubila* nudivirus (NC_040699.1), *Komagataeibacter nataicola* (NZ_CP019875.1), *Lactobacillus brevis* (NC_008497.1), *Lactobacillus plantarum* (NC_004567.2), *Plasmodium yoelii* (LR129838.1), *Saccharomyces* (NC_001133), *Tomelloso* virus (KY457233.1), *Wolbachia pipientis* (NC_002978.6)). Finally reads were mapped to *Drosophila melanogaster* genome release 6.12, and readtagger v0.4.11 was used to tag reads. Duplicate reads were marked using Picard MarkDuplicates (v2.7.1; http://broadinstitute.github.io/picard/).

Creating a panel of normals (PON) with Mutect2 (v4.1.2)

For each normal sample we ran Mutect2 with the `--disable-read-filter MateOnSameContigOrNoMappedMateReadFilter` option. We then combined these calls into a PON using CreateSomaticPanelOfNormals from the GATK suite of tools (v4.1.2; (McKenna et al. 2010).

Mutect2 (v4.1.2)

For each tumour normal pair, we ran Mutect2 specifying the unmappable regions of the genome as `--exclude-intervals`, and we used the `--dont-use-soft-clipped-bases` option to remove SNV calling in noisy reads. We specified the PON with `--panel-of-normals`. We then further filtered calls using the `FilterMutectCalls` function.

Varscan2 (v2.4)

For each tumour normal pair, we ran Varscan2 in somatic mode from combined pileups generated from both bam files using `--mpileup 1`. We specified minimum coverage of 20 for both the tumour and normal sample using `--min-coverage-normal 20 --min-coverage-tumor 20` and provided the tumour purity value estimated from Control-FREEC using `--tumor-purity`. We applied a strand filter using `--strand-filter 1`. We then ran `processSomatic` to separate high-confidence somatic SNV and indel calls. Next, we excluded high-quality somatic calls within the unmappable genome and removed sites that were also found in the PON.

Strelka (v2.9.10)

For each tumour normal pair, we ran `configureStrelkaSomaticWorkflow.py` followed by `runWorkflow.py -m local -j 4 -g 5`.

SomaticSniper (v1.0.5.0)

For each tumour normal pair, we ran `bam-somaticsniper -Q 40 -L`.

SomaticSeq (v3.3.0)

To combine calls made by multiple callers, for each tumour normal pair we ran `somaticseq_parallel.py --algorithm ada` specifying the mappable genome as ` --inclusion-region`.

FreeBayes (v1.2.0-dirty)

For each tumour normal pair, we ran `freebayes -0 --pooled-discrete --genotype-qualities`, specifying a minimum coverage of 20 with `--min-coverage 20`. Next, we excluded calls within the unmappable genome, and processed calls using the vcflib function `vcfallelicprimitives` (https://github.com/vcflib/vcflib) and vt (https://github.com/atks/vt) using `vt

decompose_blocksub` to decompose biallelic block substitutions into their constituent SNVs. We then normalised calls using `vt normalize -q`, and selected somatic calls using the vcflib function `vcfsamplediff`. We then filtered somatic calls using the vcflib function `vcffilter` to select for those with a depth greater than 20 using `vcffilter -f "DP > 20"`, for high-quality calls `vcffilter -f "QUAL > 1 & QUAL / AO > 10"` and for those supported by reads on both DNA strands `vcffilter -f "SAF > 0 & SAR > 0"` and for those with both right- and left-facing read support `vcffilter -f "RPR > 0 & RPL > 0"`.

Combining and filtering point mutations

We then merged the somatic calls from FreeBayes with the output of SomaticSeq and re-filtered the merged per-sample calls against the PON using https://github.com/bardin-lab/mutationProfiles. Combined calls were then filtered against the PON to remove germline variants and select for variants called at regions with read-depth >= 20 in both the tumour and normal sample.

Point mutation annotation

Initially, we annotated point mutations with gene information using SnpEff v4.3 (Cingolani et al. 2012), and used ISC-specific RNA-seq data (Dutta et al. 2015) to add expression levels to point mutations in genes. We then used the R package dNdScv (Martincorena et al. 2017) to annotate protein-coding mutations for their functional impact.

Downstream analysis

Downstream analyses were performed using functions developed within the mutationProfiles suite of tools (https://github.com/bardin-lab/mutationProfiles) and are detailed in our pipeline (https://github.com/bardin-lab/Riddiford_et_al_2020).

CNV-Seq

We first created a file containing read counts for each bam file using `samtools view ${bam_file} | perl -lane 'print "$F[2]\t$F[3]"' > ${out}.hits`. We then ran CNV-Seq on tumour normal pairs using window sizes of 500 bps and 50 kb with `cnv-seq.pl --window-size $window --genome-size 137547960 --global-normalization`. For small window sizes we then filtered windows with fewer than 50 reads in the normal sample.

Control-FREEC (v11.0)

For each tumour normal pair we ran Control-FREEC as described in the documentation, and annotated significant CNVs using `assess_significance.R`. We then filtered for significant CNVs, and generated .gff3 files for inspection in IGV. We further filtered CNV calls to exclude those that overlapped more than 25% with unmappable regions of the genome using bedtools subtract v2.28.0 (Quinlan and Hall 2010).

novoBreak (v1.1)

We ran novoBreak for each tumour normal pair using the recommended filtering steps.

LUMPY (v0.2.13)

For each tumour normal pair, we first extracted discordant and split reads and estimated the insert size distribution as described (https://github.com/arq5x/lumpy-sv). We then ran LUMPY using paired-end and split-reads and excluding the unmappable genome using `-x`.

svTyper (svtools 0.3.0)

A panel of normals (PON) was created using svTyper run on all normal samples and used this to genotype calls made for each tumour normal pair.

DELLY (v0.7.8)

For each tumour normal pair, we ran `delly call`, excluding unmappable regions using the `-x` option. We then ran `delly filter -f somatic` to select for somatic calls and then genotyped calls against the PON generated by svTools by running `delly call` on the output generated in the above step. We then post-filtered for somatic calls after genotyping.

Structural variant filtering and annotation

In order to filter, annotate and combine calls made from the different approaches described above, we developed a suite of tools, svParser (https://github.com/bardin-lab/svParser). By inspecting the evidence behind individual calls using `perl script/svParse.pl -i`, and viewing evidence in IGV, we optimised a set of filters that appeared to be false positives while retaining calls that were well supported. We developed a wrapper script to run our pipeline `runParser -fmas`, that runs `svParse` on per-sample variant calls made by LUMPY, novoBreak and DELLY with the filters `-f chr=1 -f su=3 -f dp=10 -f sq=0.1 -f rdr=0.05 -s`. We use the `-c` option to specify a directory containing CNV-Seq count files for each tumour normal pair, which we use to annotate average Log_2(FC) values over CNV regions. After annotating breakpoints in genes for the gene name and gene feature we inspected variants in IGV, and excluded several events that appeared to be false positive duplications called due to inconsistent coverage between the tumour and normal samples that was not consistent with a duplication in the tumour sample (Supplemental Table S2). We then combined all calls per sample, clustering variants called by multiple approaches into the same mutational event. We then annotated breakpoints in genes for expression levels using ISC-specific RNA-seq data (Dutta et al. 2015). In order to assign a putative underlying mechanism to each structural variant, breakpoints that were supported by multiple split-reads were annotated with potential microhomology sequences using SplitVision (Nazaryan-Petersen et al. 2018). Next, we attempted to re-align any sequences inserted at breakpoint junctions to the sequence flanking each breakpoint to determine whether they were locally templated, and categorised variants

according to putative underlying mechanism using criteria largely adapted from previous studies (Yang et al. 2013; Kidd et al. 2010) (Supplemental Fig. S1B).

Re-calibrating structural variant breakpoints

In order to standardise calls made between different approaches, and refine breakpoint calls, we developed svSupport (https://github.com/bardin-lab/svSupport). First, for each variant with split-read support, we extracted a breakpoint signature to classify events as deletions, tandem duplications, inversions or translocations. For each breakpoint, we searched for reads that are tagged as mapping to a non-*Drosophila* Chromosome, and filtered variants with inconsistent support between breakpoints, or where we couldn't find any split-read support after duplicate removal. We also annotated breakpoints associated with TE-tagged reads with the number of supporting reads and the TE class. Variants with low read support (< 3 split-reads) were recorded and later removed. We then incorporated tumour purity values estimated by Control-FREEC to adjust the allele frequency of each variant. Here, we first extracted the number of reads directly supporting or opposing a given breakpoint and then calculated an adjusted opposing read count given the purity of the sample as follows: `expected_oppose = (1 - tumour_purity) * total_reads`. We then used this adjusted opposing read count to calculate a tumour purity adjusted allele frequency. For example, a variant supported by 75 reads, and opposed by 25 reads with a sample tumour purity value of 0.75 would have an initial VAF of 0.75 (75/(75 + 25)), and an adjusted VAF of 1 (supporting/(supporting + (opposing - (1 - purity) * total reads)); 75/(75 + (25 - (1 - 0.75) * (75 + 25))). This step enabled us to better estimate the timing of mutations. For variants with imprecise breakpoints (CN events called by read-depth-based approaches) we first counted the number of reads mapped in the CNV region in both the tumour and normal sample. Next, we normalised read counts for sequencing depth by counting the number of reads mapped across all full Chromosomes (2L, 2R, 3L, 3R, 4, X and Y). We then performed a similar adjustment as described above for split-read-supported variants to adjust allele frequency. Here, the supporting reads were derived by subtracting the

read count in the tumour sample from the read count in the normal sample. The opposing reads were then calculated by subtracting the supporting read count from the normal read count, and a tumour purity-adjusted allele frequency calculated as above.

Clustering breakpoints

In order to identify complex events from clusters of linked breakpoints, we used a tool 'svStitch' developed in the svParser suite (https://github.com/bardin-lab/svParser). For each sample we search for variants whose breakpoints were within a 5 kb window. In cases where clustered variants all belonged to the same class of CN event (deletion or duplication) individual variants were collapsed into a single event and the variant type was not modified. In cases where multiple classes of structural variant class were clustered, or all classes belonged to the same class but were not CN events, we re-annotated each variant type in the cluster as "complex", and collapsed variants into a single mutational event.

Annotating CNVs with SNP frequencies

As a final filtering step, we used germline SNP calls made by FreeBayes to remove spurious CNV calls. First, allele frequencies of heterozygous germline SNPs called at sites with a depth > 20 in both the tumour and normal sample were extracted over a CN region. For each SNP, we calculated the difference in frequency between the tumour and normal sample, and recorded the difference as CNV-supporting if it was > 10%. We recorded the number of informative SNPs over CNVs, and in cases where we had sufficient evidence to reject a CN call (> 5 informative SNPs and supporting SNPs / opposing SNPs < 2) we marked variants as false positives. Importantly, considering small CN events are less likely to harbour enough informative SNPs for this filter to be applied, our pipeline is unable to exclude such events on this basis.

Downstream analysis

Downstream analyses were performed using functions developed within the svBreaks suite of tools (https://github.com/bardin-lab/svBreaks), and are detailed in our pipeline (https://github.com/bardin-lab/Riddiford_et_al_2020).

Identification of *Tomelloso* virus at breakpoint junctions and in sequencing data

In one sample (P31) we detected reads at a breakpoint junction in *Notch* whose mates were not mapped to the *Drosophila* genome. In order to detect the source of these unmapped reads, we extracted unmapped reads, and assembled them into contigs using CAP3 (Huang and Madan 1999). We then used the assembled contigs to query the nr database using BLASTn (Altschul et al. 1990), and identified dsDNA nudivirus *Tomelloso* (Palmer et al. 2018) as the source. We then extended this search genome-wide, by identifying somatic clusters of reads with unmapped mates and assembling their mates as described above, but found no other instances of viral integration in other samples. In order to assess the extent to which *Tomelloso* was present in our sequencing data, we combined the *Drosophila* and *Tomelloso* genomes tagged reads that likely originated from *Tomelloso* using readtagger v0.4.11. We then analysed the percentage of paired-end reads that mapped with high confidence to the *Tomelloso* genome in all samples, and performed a Pearson correlation to determine the relationship between mutation per-sample mutation count and viral load. These analyses excluded a relationship between viral load and mutation frequency.

Genotyping sequenced samples

To estimate the genotype similarity between sequenced samples, we used FreeBayes v1.2.0-dirty to call SNPs in all tumour and normal samples. We then intersected SNPs found in all samples and recorded the number of samples that shared any given SNP, and genotyped each SNP as follows: “germline recurrent” - SNPs found in the tumour and its paired head as

well as N other samples; “germline private” - SNPs found only in the tumour and its paired head.

Pipeline validation against simulated tumour genomes

In order to validate the performance of our pipeline in identifying known structural variants we used VISOR v1.0 to simulate breakpoints across the *Drosophila* genome (simulated input in Supplemental Table S3). First, we generated bed files containing 90 structural variants belonging to different classes (deletion, tandem duplication and inversion) that were distributed throughout the genome. These were used to generate simulated tumour genomes using VISOR HACk. We also generated ‘normal’ genomes containing no structural variants for comparison. We then simulated sequencing data generated from these genomes at five different levels of normal-in-tumour contamination, representing tumour purity values of 100%, 80%, 60%, 40% and 20%, and at two different sequencing depths representing a ‘low’ average coverage (tumour: 10x, normal 30x) and a ‘high’ average coverage (tumour: 30x, normal: 50x). Resulting .bam files were then processed as described for our tumour samples, and we categorised the variants we detected into true positives, false positives and false negatives according to the ground truth of SVs we simulated. Our pipeline recovered a high number of these simulated variants (93.7% in high purity, high coverage condition), and detected very few false positives (comprising 1.4% of call (Supplemental Fig. S2; Supplemental Table S4).

Pipeline validation against (Chakraborty et al. 2019) **data**

To validate our pipeline against a set of known calls, we took advantage of an existing *de novo* assembly of the non-reference *Drosophila melanogaster* strain and the annotation of its SVs previously described (strain “A4” in Chakraborty et al. 2019). We used wgsim (https://github.com/lh3/wgsim) to generate 40M 150 bps paired-end reads from this genome. In order to construct a “tumour” sample, we then combined these reads with 40M 150 bps

reads generated from the *Drosophila melanogaster* reference genome (6.12). We then generated 80M reads from the *Drosophila* reference genome as a "normal" sample. Samples were processed as a tumour normal pair using our pipeline.

We then downloaded the raw, unfiltered structural variant calls for the A4 genome characterised in Chakraborty et al. 2019 (https://github.com/mahulchak/dspr-asm/blob/master/variants-raw/sv.a4.txt.gz). To better compare SV calls between pipelines we filtered these calls to select for SVs >= 100 bps and removed calls that overlapped (> 10%) with unmappable genome. We then used the linux command shuff to randomly select 50 of this subset of unfiltered Chakraborty A4 genome SVs (Supplemental Table S5), as well as 50 of the variants called by our pipeline (Supplemental Table S6), for manual characterisation in IGV. Our pipeline correctly identified 16/50 (32%; Supplemental Table S5) variants in the A4 genome. Manual inspection validated that these were "true positives" in both Riddiford and Chakraborty. In one case (representing 2% of calls) we detected TE involvement, and scored this event as "NA". In 32/50 calls (64%; Supplemental Table S5), our pipeline did not detect SVs at the genomic loci found in Chakraborty et al. Manual inspection of these calls revealed them to be likely false positive calls in pre-filtered calls from Chakraborty et al. 2019. Therefore, 18 (50 – 32; 36%) of the Chakraborty were "true" calls, of which our pipeline identified 16. This true positive rate of 90% (16/18) is in agreement with the true positive rate we detected in our simulated analysis (93.7%).

We also performed the same analysis using the SV calls present in the UCSC genome browser session referenced in Chakraborty et al. 2019 (http://goo.gl/LLpoNH; SV calls: http://wfitch.bio.uci.edu/~tdlong/SantaCruzTracks/DSPR_R6/dm6/variation/SV.0328.vcf.gz). Here, we note that 9/32 (28.1%) of the false positive calls in the A4 genome call set were not present in the group-wise calls (scored as "NA" in Supplemental Table S5). It is possible in these cases an extra filtering step that was not detailed in the manuscript was performed, which could artificially increase the false positive rate we detect in their calls. We also note that the remaining (non-"NA") Chakraborty false positive calls were classed as ":COM" events; which the authors considered unreliable (J.J. Emerson, pers. comm.) Altogether, these analyses further validate the accuracy of our pipeline to detect known structural variants.

In order to assess the overlap between the SVs identified by our pipeline those of A4 genome of Chakraborty et al, 2019, we randomly selected 50 SVs variants called by our pipeline. Overall, we found that 41/50 (82%; Supplemental Table S6) of the SVs called by our pipeline were also present in raw calls of the A4 genome of Chakraborty et al and upon visual inspection were deemed "true positives". In addition, in 8% of our calls, we found clear support for SVs that were not present in the raw calls from the A4 genome of Chakraborty et al, but upon visual inspection were clearly true positives. Therefore, under these conditions, we consider our true positive rate to be 90%. We note the similarity of this rate to the true positive rate (93.7%) that we detected for high-depth sequencing conditions at tumour purity values of 1 in our simulated data approach (Supplemental Tables S3, S4).

Pipeline validation against using whole-gut sequencing data

As a second approach to validating our pipeline and to specifically address error rates associated with real sequencing data generated in our lab, we analysed the genomes of single whole-guts dissected from both young and aged male flies compared to the head of the same individual (http://www.ebi.ac.uk/ena/data/view/PRJEB44312). For sequencing of non-neoplastic tissues, we isolated single guts or heads from 1-week or 6-7-weeks-old male Pros>2xGFP flies (4 individual flies for each time-point). Guts were visually inspected to exclude samples with any GFP accumulation suggesting a presence of neoplasia. Tissues were dissected in ice-cold, nuclease free PBS and DNA was isolated immediately with the QIAamp DNA micro kit (Qiagen) according to manufacturer instructions. gDNA was eluted with nuclease-free water and DNA quantity was measured with the Qubit dsDNA Broad Range Assay Kit (Thermo Fisher Scientific). We then used 0.6 ng of purified gDNA for library preparation with the Nextera XT protocol (Illumina) and whole-genome 2 x 150 bps paired-end sequencing was performed on NovaSeq (Illumina).

These whole guts should not contain tumours and therefore should not have large clonal expansions as the ones that we used for somatic mutation detection. If somatic

mutations were present in these samples, they are predicted to have a much lower allele frequency than those expanded by somatic mutation of *Notch* in vivo. In young individuals, where we would expect to detect very low levels of mutation, we detected significantly lower levels of each class of mutation than with our tumour samples, supporting the low false-positive levels we found in our simulated data (Supplemental Fig. 8A). The low level of events could be true somatic events or could be false positives. However, this analysis puts an upper-bound on the putative false positive rate detected in real sequencing data, and further validates the low false positive levels found in our simulated data approach.

We then analysed the aged whole-guts as a developmentally closer comparison tissue to our tumour samples. In these samples, we detected substantially higher levels of mutations than in the young samples (Supplemental Fig. 8B). Interestingly, we found that duplications comprised the large majority of SVs detected in the aged whole-gut samples (81%; Supplemental Fig. 8C), and that these were mostly < 5kb (Supplemental Fig. 8D). Although it is possible that this tissue harboured developmental mutations that were detected in stem cell lineages, we believe that this probably constitutes a false positive signature that our pipeline fails to remove. This is likely explained by a novel filtering step that we developed to annotate CN events with heterozygous SNP allele frequencies: We require a minimum number of informative SNPs (heterozygous; high quality; depth > 20) to mark a CN event as a false positive, and through the analysis in Supplemental Fig. 8C, we conclude that small CN events do not harbour enough SNPs to be removed by this filter. While we did not detect many SVs with such a signature in our tumour SV calls (Supplemental Fig. 8D), and therefore it is unlikely that this influenced our data set, we note that our pipeline is unable to exclude small SV events on this basis.

Additionally, we detected a high number of indels called in the aged whole-gut samples (Supplemental Fig. 8B). These were largely very low allele frequency events (Supplemental Fig. 8E; cell fraction <= 0.1). Again, comparing with the indel calls we found in our tumour

samples (Supplemental Fig. 8E), we did not detect many indels with similar allele frequencies (<= 0.1), suggesting that it was not a major source of false positives in our analysis.

Genome feature discovery

To identify regions of the *Drosophila* genome release 6.12 susceptible to forming non-B-form DNA structures we first scanned the genome for inverted repeat sequences following the approach outlined in (Zou et al. 2017), as well as for G-quadruplexes using the R package G4Hunter (Bedrat et al. 2016). To identify short sequence motifs in *Notch* breakpoint regions, we used MEME (Bailey and Elkan 1994), using sequences extracted +/- 500 bps from each breakpoint in *Notch* as input. We then used FIMO (Grant et al. 2011) to search for and annotate recovered motifs genome-wide.

Association of mutations with genomic regions

To detect the enrichment or depletion of genomic regions for mutations we counted the number of mutations in a given region, and compared this to the expected number considering the region's size. The association was tested by performing a two-sided binomial test, adjusting for multiple comparisons using Benjamini-Hochberg adjustment. We require that the number of observed hits + expected hits is greater than 10 for plotting. To assess whether breakpoints in *Notch* were enriched for poly(dA:dT) sequences, the sequence +/- 500 bps around each breakpoint in *Notch* was extracted and permutation tests were performed using regioneR (Gel et al. 2016) on the overlap between repeats and these breakpoint-flanking regions. In order to compare observed counts between real and shuffled data, we restricted permutations to within the genomic locus X:2700000-3400000, and performed 10,000 permutations. To test whether mutations were found closer to poly(dA:dT) tracts than expected by chance, all classes of mutation (structural variant breakpoints, SNV and indels) were combined. We then simulated 10 times as many mutations as we found in our combined calls across the mappable genome with distribution across chromosomes equal to that observed in

our combined mutation data to act as a comparison. Finally, we calculated the relative distances of both mutations and simulated data to the closest repeat sequence using bedtools reldist (Favorov et al. 2012), and performed a Kolmogorov–Smirnov test to compare distributions.

Immunofluorescence

Dissection of a tumour (Supplemental Fig. S7A) marked by accumulation of GFP+ cells (ISCs) in Dl-Gal4>UAS-GFP flies and immunofluorescence to mark GFP, enteroendocrine cells (Pros+) and DAPI, was done as previously published (Siudeja, Cell Stem Cell, 2015).

Log 2 Fold Change Plots

The log 2 Fold Change in coverage across the genome was plotted for Supplemental Figure S8B and C.

References

Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. 1990. Basic local alignment search tool. *J Mol Biol* **215**: 403–410.

Bailey TL, Elkan C. 1994. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proc Int Conf Intell Syst Mol Biol* **2**: 28–36.

Bedrat A, Lacroix L, Mergny J-L. 2016. Re-evaluation of G-quadruplex propensity with G4Hunter. *Nucleic Acids Res* **44**: 1746–1759.

Boeva V, Popova T, Bleakley K, Chiche P, Cappo J, Schleiermacher G, Janoueix-Lerosey I, Delattre O, Barillot E. 2012. Control-FREEC: a tool for assessing copy number and allelic content using next-generation sequencing data. *Bioinformatics* **28**: 423–425.

Chakraborty M, Emerson JJ, Macdonald SJ, Long AD. 2019. Structural variants exhibit widespread allelic heterogeneity and shape variation in complex traits. *Nat Commun* **10**: 4872.

Chong Z, Ruan J, Gao M, Zhou W, Chen T, Fan X, Ding L, Lee AY, Boutros P, Chen J, et al. 2017. novoBreak: local assembly for breakpoint detection in cancer genomes. *Nat Methods* **14**: 65–67.

Cingolani P, Platts A, Wang LL, Coon M, Nguyen T, Wang L, Land SJ, Lu X, Ruden DM. 2012. A program for annotating and predicting the effects of single nucleotide polymorphisms, SnpEff: SNPs in the genome of Drosophila melanogaster strain w1118; iso-2; iso-3. *Fly* **6**: 80–92.

Dutta D, Dobson AJ, Houtz PL, Gläßer C, Revah J, Korzelius J, Patel PH, Edgar BA, Buchon N. 2015. Regional Cell-Specific Transcriptome Mapping Reveals Regulatory Complexity in the Adult Drosophila Midgut. *Cell Rep* **12**: 346–358.

Favorov A, Mularoni L, Cope LM, Medvedeva Y, Mironov AA, Makeev VJ, Wheelan SJ. 2012. Exploring massive, genome scale datasets with the GenometriCorr package. *PLoS Comput Biol* **8**: e1002529.

Gel B, Díez-Villanueva A, Serra E, Buschbeck M, Peinado MA, Malinverni R. 2016. regioneR: an R/Bioconductor package for the association analysis of genomic regions based on permutation tests. *Bioinformatics* **32**: 289–291.

Grant CE, Bailey TL, Noble WS. 2011. FIMO: scanning for occurrences of a given motif. *Bioinformatics* **27**: 1017–1018.

Huang X, Madan A. 1999. CAP3: A DNA sequence assembly program. *Genome Res* **9**: 868–877.

Kidd JM, Graves T, Newman TL, Fulton R, Hayden HS, Malig M, Kallicki J, Kaul R, Wilson RK, Eichler EE. 2010. A human genome structural variation sequencing resource reveals insights into mutational mechanisms. *Cell* **143**: 837–847.

Layer RM, Chiang C, Quinlan AR, Hall IM. 2014. LUMPY: a probabilistic framework for structural variant discovery. *Genome Biol* **15**: R84.

Martincorena I, Raine KM, Gerstung M, Dawson KJ, Haase K, Van Loo P, Davies H, Stratton MR, Campbell PJ. 2017. Universal Patterns of Selection in Cancer and Somatic

Tissues. *Cell* **171**: 1029–1041.e21.

McKenna A, Hanna M, Banks E, Sivachenko A, Cibulskis K, Kernytsky A, Garimella K, Altshuler D, Gabriel S, Daly M, et al. 2010. The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res* **20**: 1297–1303.

Nazaryan-Petersen L, Eisfeldt J, Pettersson M, Lundin J, Nilsson D, Wincent J, Lieden A, Lovmar L, Ottosson J, Gacic J, et al. 2018. Replicative and non-replicative mechanisms in the formation of clustered CNVs are indicated by whole genome characterization. *PLoS Genet* **14**: e1007780.

Palmer WH, Medd NC, Beard PM, Obbard DJ. 2018. Isolation of a natural DNA virus of Drosophila melanogaster, and characterisation of host resistance and immune responses. *PLoS Pathog* **14**: e1007050.

Quinlan AR, Hall IM. 2010. BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics* **26**: 841–842.

Rausch T, Zichner T, Schlattl A, Stütz AM, Benes V, Korbel JO. 2012. DELLY: structural variant discovery by integrated paired-end and split-read analysis. *Bioinformatics* **28**: i333–i339.

Siudeja K, van den Beek M, Riddiford N, Boumard B, Wurmser A, Stefanutti M, Lameiras S, Bardin AJ. 2021. Unraveling the features of somatic transposition in the Drosophila intestine. EMBO J e106388.

Spradling A. 2017. Polytene chromosome structure and somatic genome instability. Cold Spring Harb Symp Quant Biol; 82:293-304.

Xie C, Tammi MT. 2009. CNV-seq, a new method to detect copy number variation using high-throughput sequencing. *BMC Bioinformatics* **10**: 80.

Yang L, Luquette LJ, Gehlenborg N, Xi R, Haseley PS, Hsieh C-H, Zhang C, Ren X, Protopopov A, Chin L, et al. 2013. Diverse mechanisms of somatic structural variations in human cancer genomes. *Cell* **153**: 919–929.

Zou X, Morganella S, Glodzik D, Davies H, Li Y, Stratton MR, Nik-Zainal S. 2017. Short inverted repeats contribute to localized mutability in human somatic cells. *Nucleic Acids Res* **45**: 11213–11221.