Samplot: multi-technology structural variant sequence data visualization Supplemental Figures

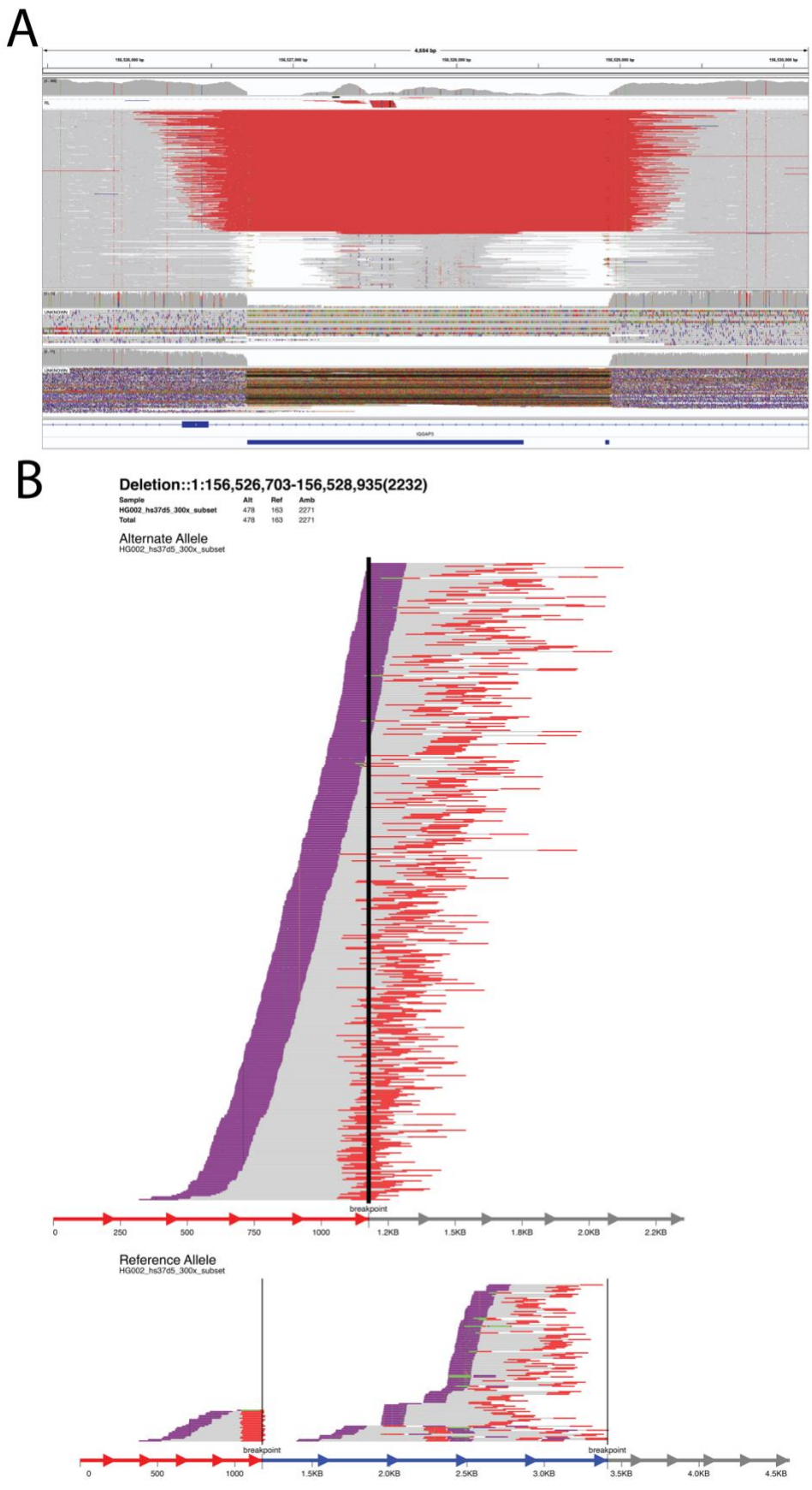
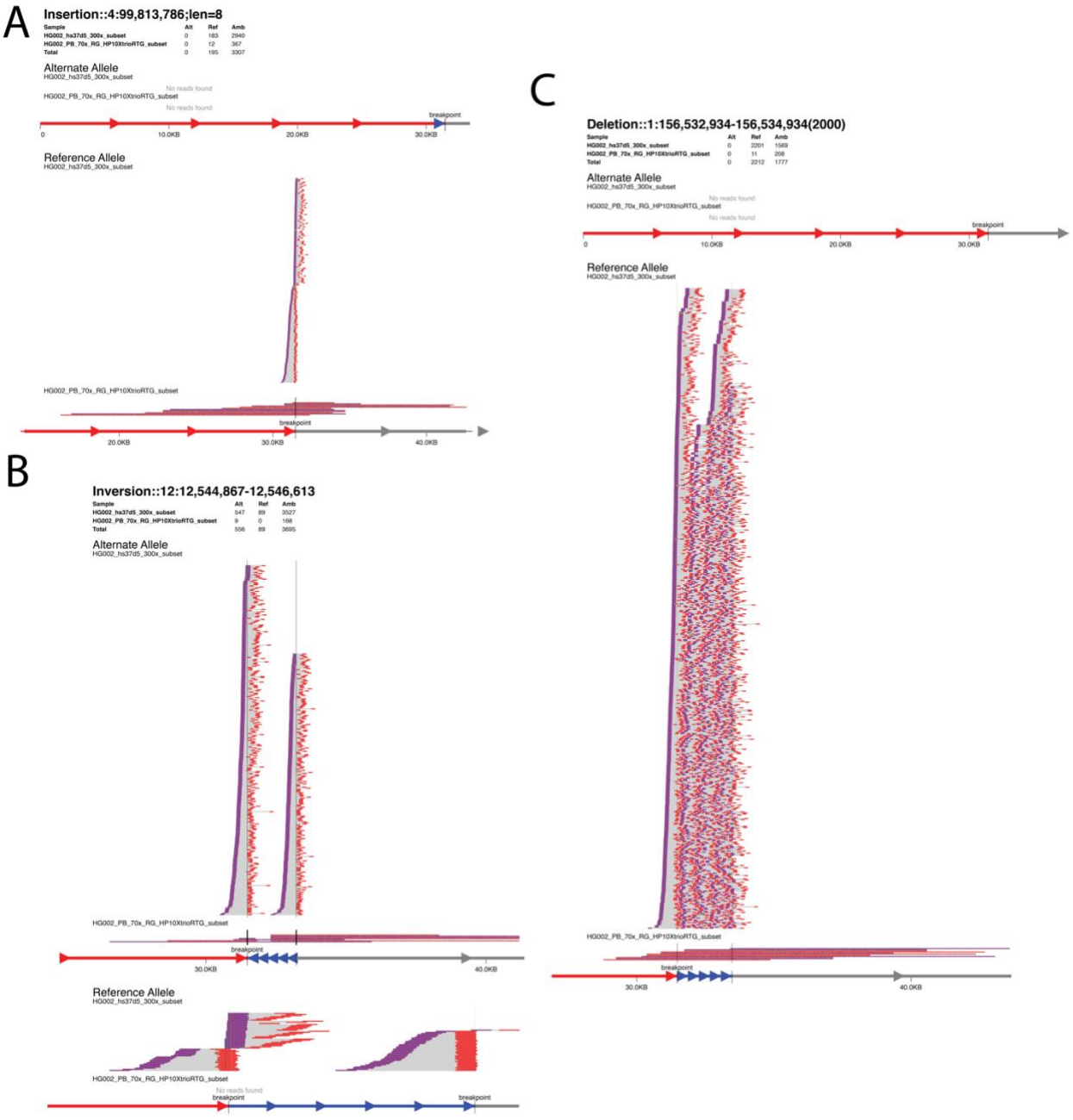
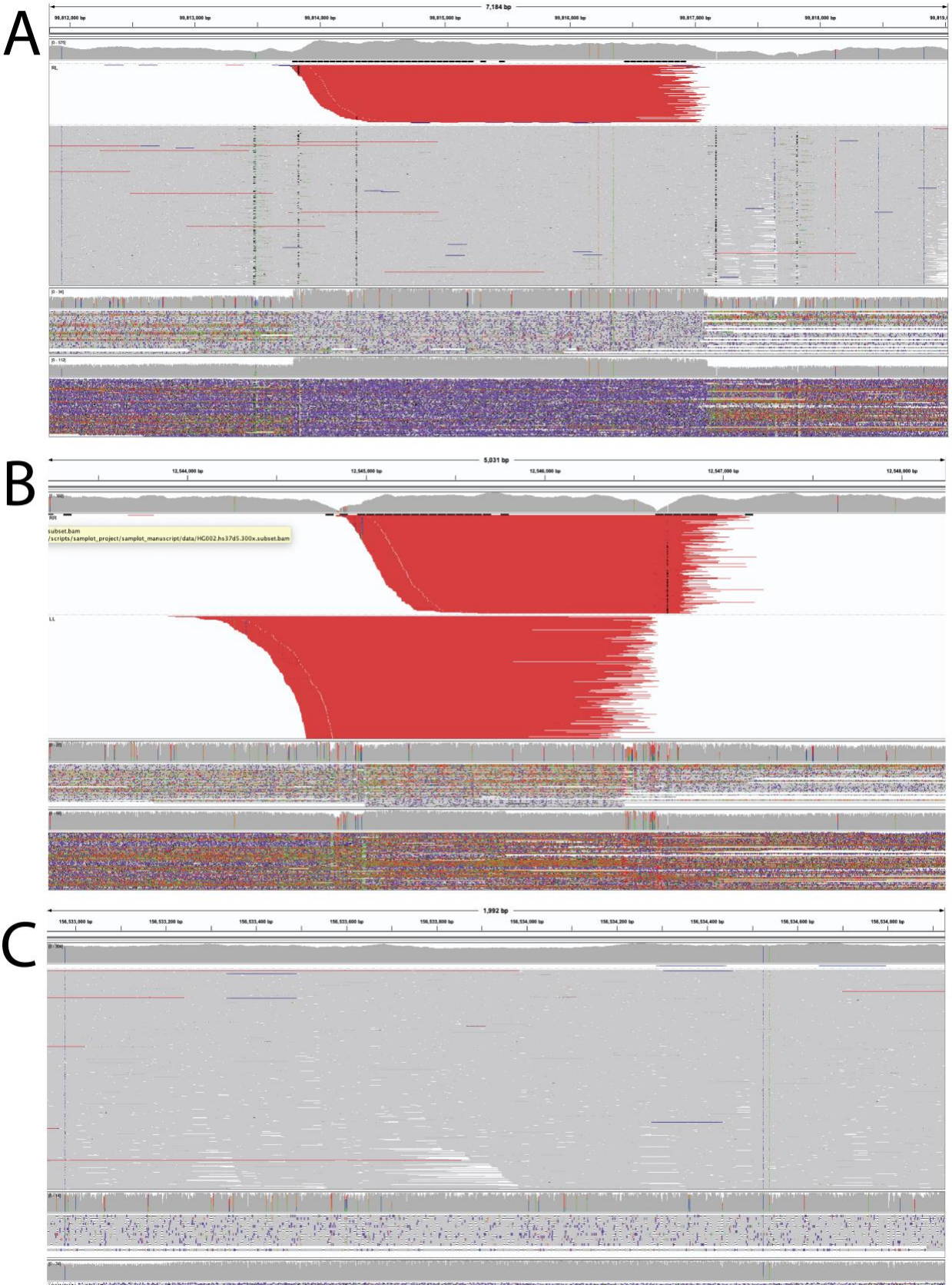


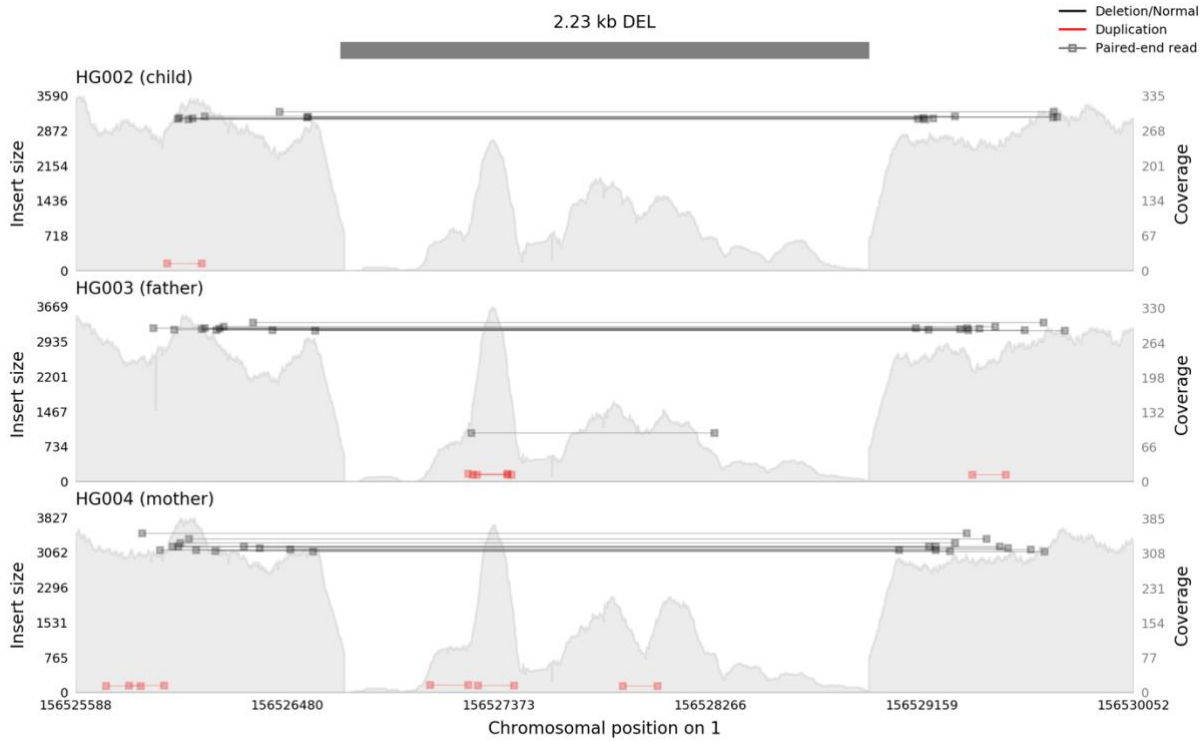
Figure 1 **Supplemental Figure 1. Integrative Genomics Viewer and svviz deletion plots**. **A)** An IGV screenshot of the same deletion variant as shown in Figure 1. Reads are shown as pairs and sorted by insert size, with coverage shown at top of image. **B)** Svviz plot of the same deletion. Reads supporting the alternate allele are shown at top, with reads supporting the reference allele at the bottom. Breakpoints are indicated by dark vertical bars.



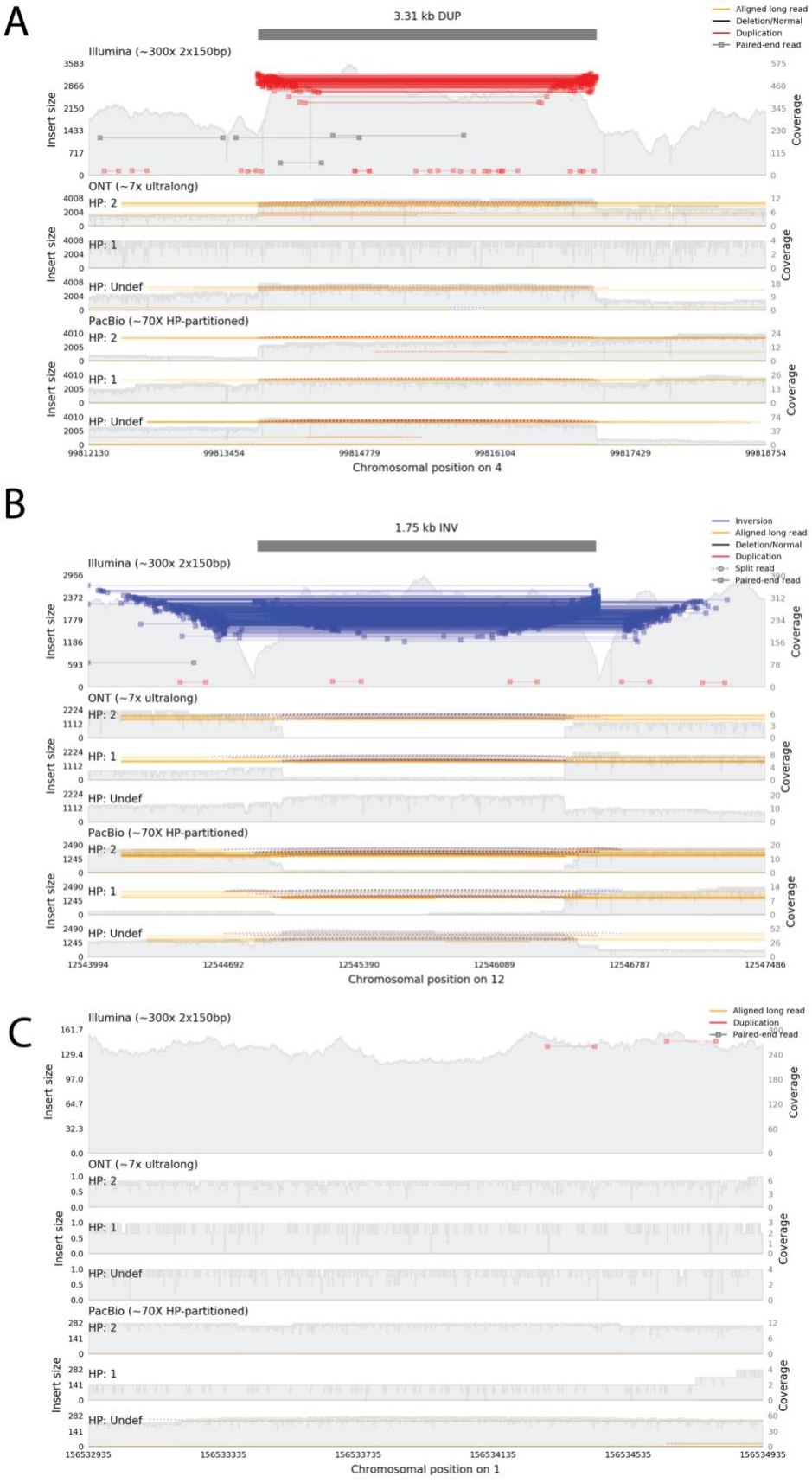
Supplemental Figure 2. Svviz images for multiple region types. A) The duplication SV from Figure plotted with svviz. B) The inversion SV from Figure 2 plotted with svviz. **C)** A region with no SV plotted with svviz.



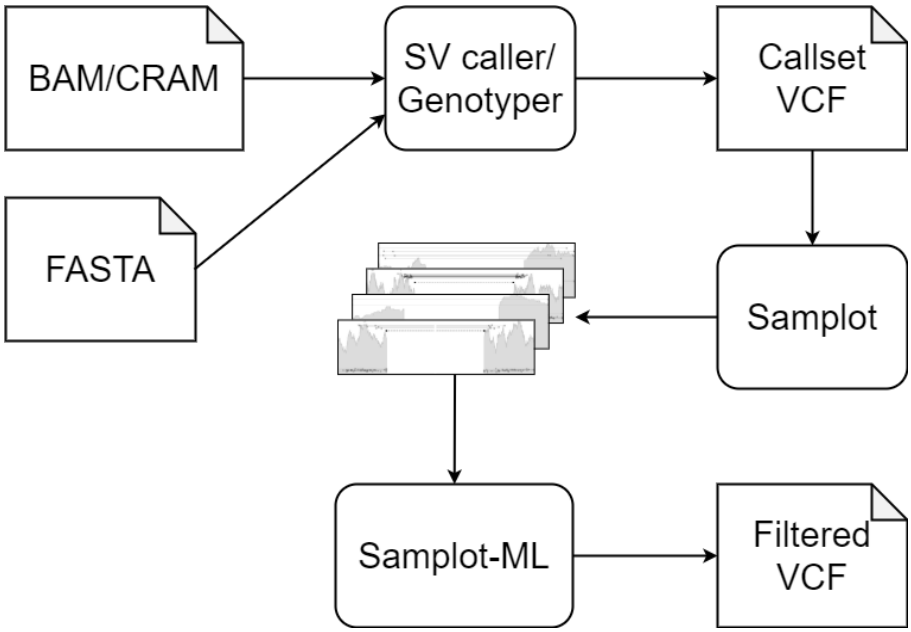
Supplemental Figure 3. IGV screenshots for multiple region types. A) The duplication SV from Figure 2 screenshot from IGV. **B)** The inversion SV from Figure 2 screenshot from IGV. **C)** A region with no SV screenshot from IGV.



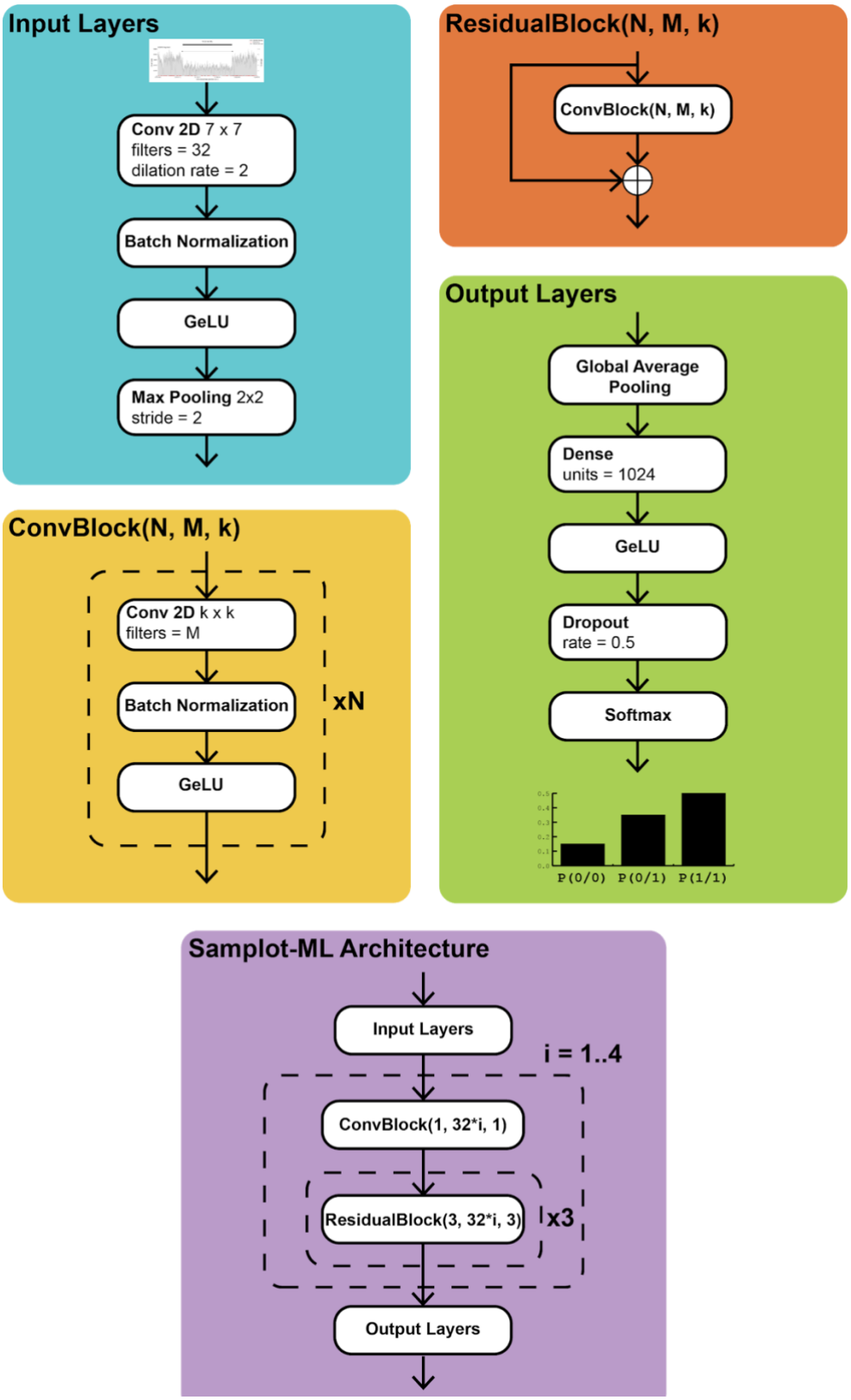
Supplemental Figure 4. A Samplot image showing a deletion variant in a trio of samples. Evidence for the variant, in the form of loss of coverage and discordant paired-end reads, appears in the child (top) and both parents.



Supplemental Figure 5. Samplot images of multiple region types with multiple sequencing technologies. A) The duplication SV from Figure 2 including Illumina, ONT, and PacBio sequence data. **B)** The inversion SV from Figure 2 including Illumina, ONT, and PacBio sequence data. **C)** A region with no SV including Illumina, ONT, and PacBio sequence data.



Supplemental Figure 5. Typical workflow for Samplot-ML



Supplemental Figure 6. Samplot-ML model architecture. GeLU refers to the Gaussian error linear unit[34]

Running Samplot-ML

From the Samplot-ML root directory:

1. Generate Images from VCF of deletion SVs

```
bcftools query -f '%CHROM\t%POS\t%INFO\END\n' $vcf_path |
gargs -p $n_processes \
      "bash data_processing/gen_img.sh \\
      --chrom {0} --start {1} --end {2} \\
      --sample $sample --genotype DEL \\
      --min-mqual 10 \\
      --fasta $fasta_reference \\
      --bam-file $bam \\
      --out-dir $out
```

gargs is an open source alternative to xargs and can be found at https://github.com/brentp/gargs

2. Crop Images

```
bash data_processing/crop.sh \
      -p $n_processes \
      -d $path_to_images \
      -o $out
```

3. Filter input VCF/modify predicted genotypes with Samplot-ML

```
find $path_to_cropped_images -name '*.png' > image-list.txt
bash evaluation/create_test_vcfs.sh \
      --model-path saved_models/samplot-ml.h5 \
      --data-list image-list.txt \
      --vcf $vcf_path \
      --num-processes $n_processes \
      --batch-size $batch_size \
      --out-dir $out
```

$n_processes is the number of cpu processes used to load images to the model. **$batch_size** is the number of images to feed to the model at once. The output directory will contain the filtered vcf and a bed file with regions and prediction scores with format:

```
Chrom    start    end    Pref    Phet    Palt
```

Where Pref, Phet, and Palt are the prediction scores for homozygous reference, heterozygous, and homozygous alternate genotypes.