BreakPointSurveyor: Supplementary Information and Methods

Structural Variant Visualization Tools

Existing SV visualization tools (Guan and Sung, 2016) include linear genome browsers (Robinson *et al.*, 2011; Sante *et al.*, 2014) which indicate breakpoint positions along a single reference genome, and tools which represent breakpoints explicitly as arcs connecting linear or circular genomic segments (Koboldt *et al.*, 2012; Krzywinski *et al.*, 2009; O'Brien *et al.*, 2010; Hiltemann *et al.*, 2013; Piazza *et al.*, 2012; Parsons, 1995).

Breakpoint Surveyor Architecture

The Breakpoint Surveyor architecture is divided into three complementary layers (Supplementary Figure 1b): the core application layer, the workflow script layer, and the project data layer. Each may be versioned and distributed individually, but for convenience the workflow and data layers may be kept together. Core applications are command-line driven programs, typically implemented in R or Python, which perform specific analysis or visualization tasks for a given sample or selected target region. Workflow scripts are shell scripts that utilize core applications to implement individual steps in the processing stages illustrated in Supplementary Figure 1a. They are generally customized according to various user-specified toolsets, locale-specific details, and the available computational infrastructure. Looping over multiple samples or regions of interest and submission of jobs to computational clusters is implemented at the workflow layer. The project data layer contains the input and output of each workflow step;

these data records are versioned and allow for reproducibility, provenance tracking, and distributed computing (Ram, 2013). Such modularity also facilitates scaling of the workflow, from experimental one-off analysis to distributed analysis on clusters.

Distribution

The entire BreakPointSurveyor package, including the Core code, workflow, and data, as well as comprehensive documentation and installation instructions, can be obtained here:

https://github.com/ding-lab/BreakPointSurveyor

The BreakPointSurveyor project provides three reference workflows, each implemented as separate git branches. These workflows are,

- **TCGA_Virus** (master branch): Comprehensive workflow and data for one TCGA virus-positive sample (TCGA-BA-4077-01B-01D-2268-08) which has been aligned to a custom reference
- **1000SV** (1000SV branch): Analysis of discordant reads on publicly available human sample
- **Synthetic** (Synthetic branch): Creation and analysis of a dataset containing an inter-chromosomal breakpoint

This Supplementary Methods document and manuscript focus primarily on the TCGA_Virus workflow. The other workflows are generally simplified versions of TCGA_Virus with publicly available data. See online instructions for details.

Data Acquisition and Preprocessing

WGS data were downloaded from CGHub (https://cghub.ucsc.edu/) and subsequently realigned to a custom reference sequence. This reference consists of the GRCh37 human reference together with genomic sequences from 18 virus types that are either known or at least suspected to be associated with cancer, including various HPV and human herpesvirus (HHV) subtypes (see Supplementary Table 1). Realignment was performed using BWA(Li and Durbin, 2009) v. 0.5.9 with parameters "-t 4 -q 5::".

Virus	**NCBI/Genbank Code**	**Virus**	**NCBI/Genbank Code**
HPV6b	NC_001355.1	HPV56	EF177177.1
HPV16	NC_001526.2	HPV58	D90400.1
HPV18	NC_001357.1	HPV59	X77858.1
HPV31	J04353.1	Polyoma BK	NC_001538.1
HPV33	M12732.1	Polyoma HPyV7	NC_014407.1
HPV35	M74117.1	HHV1	JQ780693.1
HPV39	M62849.1	HHV4 (Epstein Barr)	NC_009334.1
HPV45	EF202167.1	HHV5 (Cytomegalovirus)	AY446894.2
HPV52	X74481.1	Hepatits B	NC_003977.1

Supplementary Table 1. Custom reference for WGS realignment is based on GRCh37 with additional virus sequences. Table lists these viruses and associated NCBI accession numbers. See https://github.com/ding-lab/BreakPointSurveyor/tree/master/A_Reference for details.

Integration Detection

We employed three types of analyses to identify breakpoints: identification of discordant read pairs, insert size analysis, and contig realignment. BPS evaluated discordant reads as read pairs with mapping quality > 25 where one mate pair mapped to the human reference and the other to a viral genome. Samples were also analyzed with Pindel(Ye *et al.*, 2009) read-pair module, which analyzed insert size to localize breakpoint positions to within approximately ± 500 bp. TIGRA-SV(Chen *et al.*, 2014) was then used to assemble contigs in the vicinity of these breakpoints, with realignment of contigs to the human+virus reference yielding exact breakpoint positions.

Breakpoint Clustering and Plot List

We group multiple nearby breakpoints on the same chromosome pair into clusters to serve as preliminary integration event predictions and regions of interest for subsequent analysis and visualization. With each breakpoint between chromA and chromB represented by a point with coordinates (posA, posB), the clustering algorithm draws a square with sides length L/2 centered at each breakpoint, and all breakpoints within overlapping squares are grouped into one cluster. Nearby Pindel breakpoints between the same chromosome and virus (those occurring within 50Kbp of one another along both genomes) were clustered into integration events, which defined regions of interest for all subsequent analyses. Note that for the sample of interest all three breakpoints were nearby.

Regions of interest for the structure plots consist of the domain of the cluster plus padding of +/-50Kbp. These regions of interest are then saved to the PlotList file and drive downstream analysis and visualization stages for structure analysis.

Contig Alignment

Contig alignment improves breakpoint predictions by assembling a consensus sequence (contig) from reads spanning a breakpoint, then re-aligning the contig to the human+virus reference. Contigs are created using Tigra-SV(Chen *et al.*, 2014) based on breakpoint predictions from Pindel(Ye *et al.*, 2009) and aligned with BWA mem (v. 0.7.10) to the human+virus reference described above. The resulting SAM file is then processed to yield a series of files (described below) used for plotting and analysis, with support for multiple breakpoints per contig.

- pSBP: One entry per alignment (line) of SAM(Li *et al.*, 2009) file, describing alignment of one segment to reference
- SBP: Incorporates pair of pSBP segments, Sa and Sb, describing one breakpoint. Multiple (N) breakpoints in a contig will result in (N-1) SBP lines
- rSBP: Retains only chromosome names and breakpoint position (BPC Format)
- qSBP: Gives information about paired breakpoints (>1 breakpoints per contig). For N breakpoints per contig, N-1 qSBP entries are generated.

The rSBP file is used in the breakpoint panel of the BPS structure plot to illustrate contig predictions.

Copy Number

Copy number is evaluated by first subsampling read depth uniformly over the selected target region to obtain read depth values at approximately 10,000 positions. Average read depth is estimated for each WGS BAM file which spans the entire genome as the product of the number of mapped reads and read length divided by genome length. Copy number at a given genomic position is defined as twice the read depth at that position divided by average read depth, to yield an average copy number of 2.

Expression analysis

For the RPKM (RNA-Seq-based) analysis, we obtained a nonredundant set of merged exons from the union of all gene-specific transcripts annotated in Ensembl Release75. Raw RNA-Seq reads were counted for each exon using the BEDTools (Quinlan and Hall, 2010) utility suite. Per-exon RPKM was evaluated conventionally as (10^9 * R)/(N * L), where R is the number of raw reads mapped to an exon, N is the count of total mapped reads, and L is exon length. RPKM values were calculated for all exons of interest (defined as those within 1Mbp of an integration event) over all samples of a given disease; these calculations yield sets of controls obtained from RNA-Seq tumor samples of the same disease without integration events to support comparison of disease cases.

We also demonstrate a RSEM workflow, where an expression file downloaded from TCGA Firehose replaces the RNA-Seq analysis described above.

In both cases, exons for which more than half the cases have zero expression were excluded from analysis. In cases where an integration event overlaps a gene, we considered exons that lie upstream, downstream, and within the integration event independently. We then calculated the relative increase/decrease of expression for each gene in the case sample with respect to controls using permutation testing for the combined expression of all exons in the gene. Specifically, the test evaluates the null hypothesis that a gene with integration event in the case is not significantly dysregulated with respect to the controls. We then perform multiple test correction in the form of per-gene FDRs using the Benjamini and Hochberg method (Benjamini and Hochberg, 1995) for all genes in a single integration event.

BPC, BPR, and Depth Data Types

To support the visualization of output from various tools using a common set of functions, multiple types of SV predictions are represented in a common format. We define two simple plaintext-encoded, tab-separated (TSV) file formats to encode the precise positions or regions of breakpoints between chromosomes A and B (where chromosome B may be the virus genome): The Breakpoint Coordinate (BPC) format represents precise breakpoint position using columns [chromA, posA, chromB, posB], where chromA < chromB (or posA < posB if

chromA = chromB). An optional fifth column can be any text string encoding an attribute of the breakpoint, which will be mapped to e.g. different color in the figure. Similarly, the Breakpoint Region (BPR) format describes breakpoint regions using columns [chromA, startPosA, endPosA, chromB, startPosB, endPosB, (attribute)], again with the last column being optional.

Figure Generation

Multi-panel figures are generated using the R-language ggplot2 (Wickham, 2016) package in three steps: data processing, panel rendering, and figure assembly. The data processing step normalizes data into standard formats. For instance, breakpoint predictions from different SV callers are normalized into a BPC file format (defined below), while read depth and gene annotation are converted to Depth and BED(Quinlan and Hall, 2010) formats, respectively. Each dataset is rendered as an image panel using the ggplot() function and saved as a binary "GGP" object with saveRDS(). GGP objects can be visualized using the ggp2pdf utility. Additional layers, for instance predictions from different SV callers, may be added to an existing GGP object in a subsequent processing step with data from a different BPC (or BPR) file. Finally, multiple GGP objects are assembled, aligned to common axes, and saved to a PDF format during figure assembly to form a composite figure. Such delegation of tasks facilitates incorporation of additional tools and analyses into the workflow without modifications to the core apps.

Additional Workflows and Results

Figures for the three BreakPointSurveyor workflows are illustrated in Supplementary Figures as well as online.

TCGA Virus. Structure and expression plots for the TCGA-BA-4077-01 sample are shown in Figure 1 in the manuscript, as well as in Supplementary Figures 4 and 5 and online at https://github.com/ding-lab/BreakPointSurveyor. Expression plots for both RNA-Seq based RPKM analysis as well as the alternative TCGA RSEM-based analysis are shown. Supplementary Figures 2 and 3 illustrate structure and expression plots, respectively, for four additional TCGA samples with viral integration.

1000SV. The 1000SV workflow investigates interchromosomal human-human breakpoints in a publicly available human sample from the 1000 Genomes project, NA19240, which was sequenced at high (80X) coverage. Supplementary Figures 6 and 7 illustrate to example discordant read clusters, events AQ and AU, respectively, which illustrate distinct discordant read patterns and illustrate how visualizing discordant reads as coordinates on a plane yields patterns and interpretations which would not be discernable in other representations. Figures available online at https://github.com/ding-lab/BreakPointSurveyor/tree/1000SV

Synthetic. The Synthetic workflow generates *de novo* an interchromosomal translocation by concatenating two reference segments, then creating synthetic (simulated) reads in this region. After realigning with BWA to a custom minimal reference, this small BAM is analyzed and visualized with a standard BPS workflow. Supplementary Figure 8 illustrates discordant reads mapping to a junction between chr 9 and 22, together with copy number in that region. The structure plot also demonstrates gene and exon annotation functionality, including exon labels, with synthetic genes and exons. Figure available online at https://github.com/ding-lab/BreakPointSurveyor/tree/Synthetic/T_PlotStructure

References

Benjamini,Y. and Hochberg,Y. (1995) Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, **57**, 289–300.

Cao,S. *et al.* (2016) Divergent viral presentation among human tumors and adjacent normal tissues. *Sci. Rep.*, **6**, 28294.

Chen,K. *et al.* (2014) TIGRA: a targeted iterative graph routing assembler for breakpoint assembly. *Genome Research*, **24**, 310–317.

Guan,P. and Sung,W.-K. (2016) Structural variation detection using next-generation sequencing data: A comparative technical review. *Methods*, **102**, 36–49.

Hiltemann,S. *et al.* (2013) iFUSE: integrated fusion gene explorer. *Bioinformatics*, **29**, 1700–1701.

Koboldt,D.C. *et al.* (2012) Massively parallel sequencing approaches for characterization of structural variation. *Methods Mol. Biol.*, **838**, 369–384.

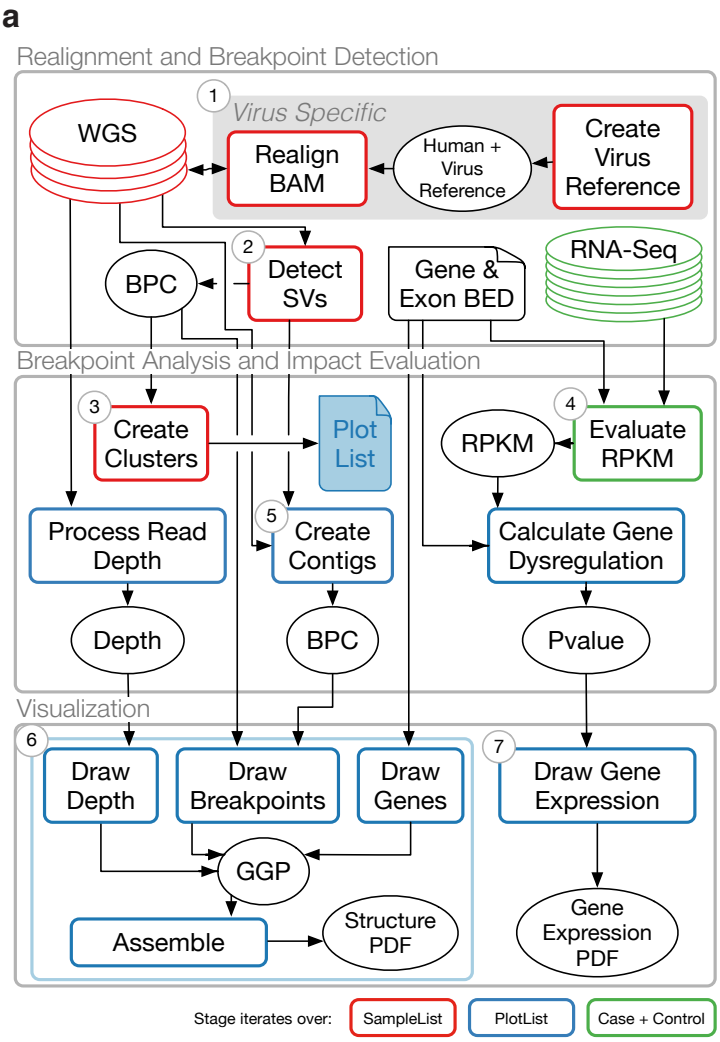
Krzywinski,M. *et al.* (2009) Circos: an information aesthetic for comparative genomics. *Genome Research*, **19**, 1639–1645.

Li,H. and Durbin,R. (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, **25**, 1754–1760.

Li,H. *et al.* (2009) The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, **25**, 2078–2079.

O'Brien,T.M. *et al.* (2010) Gremlin: an interactive visualization model for

analyzing genomic rearrangements. *IEEE Trans Vis Comput Graph*, **16**, 918–926.
Parsons,J.D. (1995) Miropeats: graphical DNA sequence comparisons. *Comput. Appl. Biosci.*, **11**, 615–619.
Piazza,R. *et al.* (2012) FusionAnalyser: a new graphical, event-driven tool for fusion rearrangements discovery. *Nucleic Acids Res.*, **40**, e123–e123.
Quinlan,A.R. and Hall,I.M. (2010) BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics*, **26**, 841–842.
Ram,K. (2013) Git can facilitate greater reproducibility and increased transparency in science. *Source Code Biol Med*, **8**, 7.
Robinson,J.T. *et al.* (2011) Integrative genomics viewer. *Nat Biotechnol*, **29**, 24–26.
Sante,T. *et al.* (2014) ViVar: A Comprehensive Platform for the Analysis and Visualization of Structural Genomic Variation. *PLoS ONE*, **9**, e113800–12.
Wickham,H. (2016) ggplot2 Springer, Cham.
Ye,K. *et al.* (2009) Pindel: a pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads. *Bioinformatics*, **25**, 2865–2871.



b

Breakpoint Surveyor Architecture

Core App

- Task specific, command-line driven programs
- Typically R or Python implementation
- Process one dataset at a time

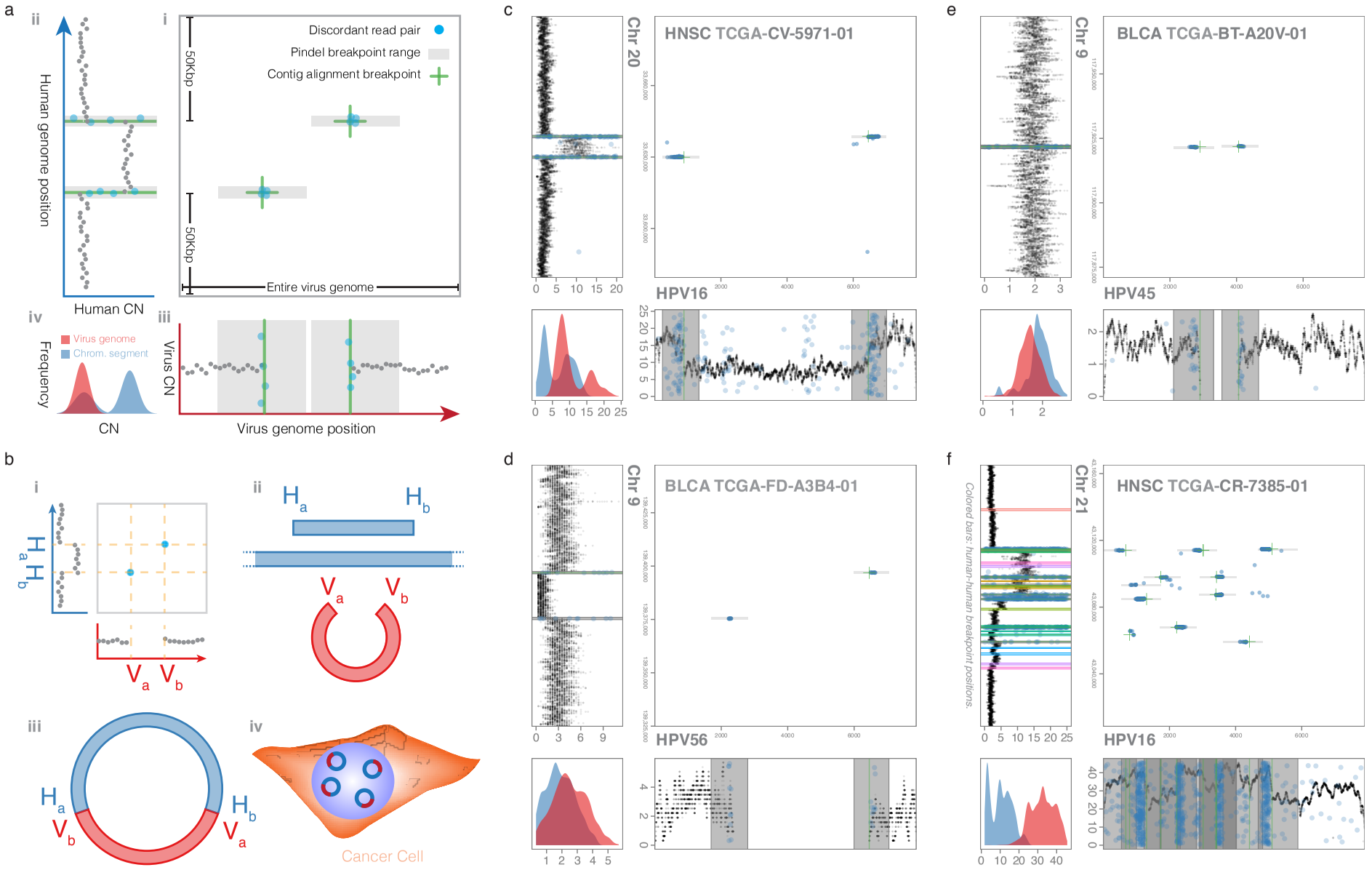
Workflow Script

- Series of shell scripts executed in defined order to implement workflow
- Modified to suit locale, toolset, and computational infrastructure
- Loop over lists of samples or plots

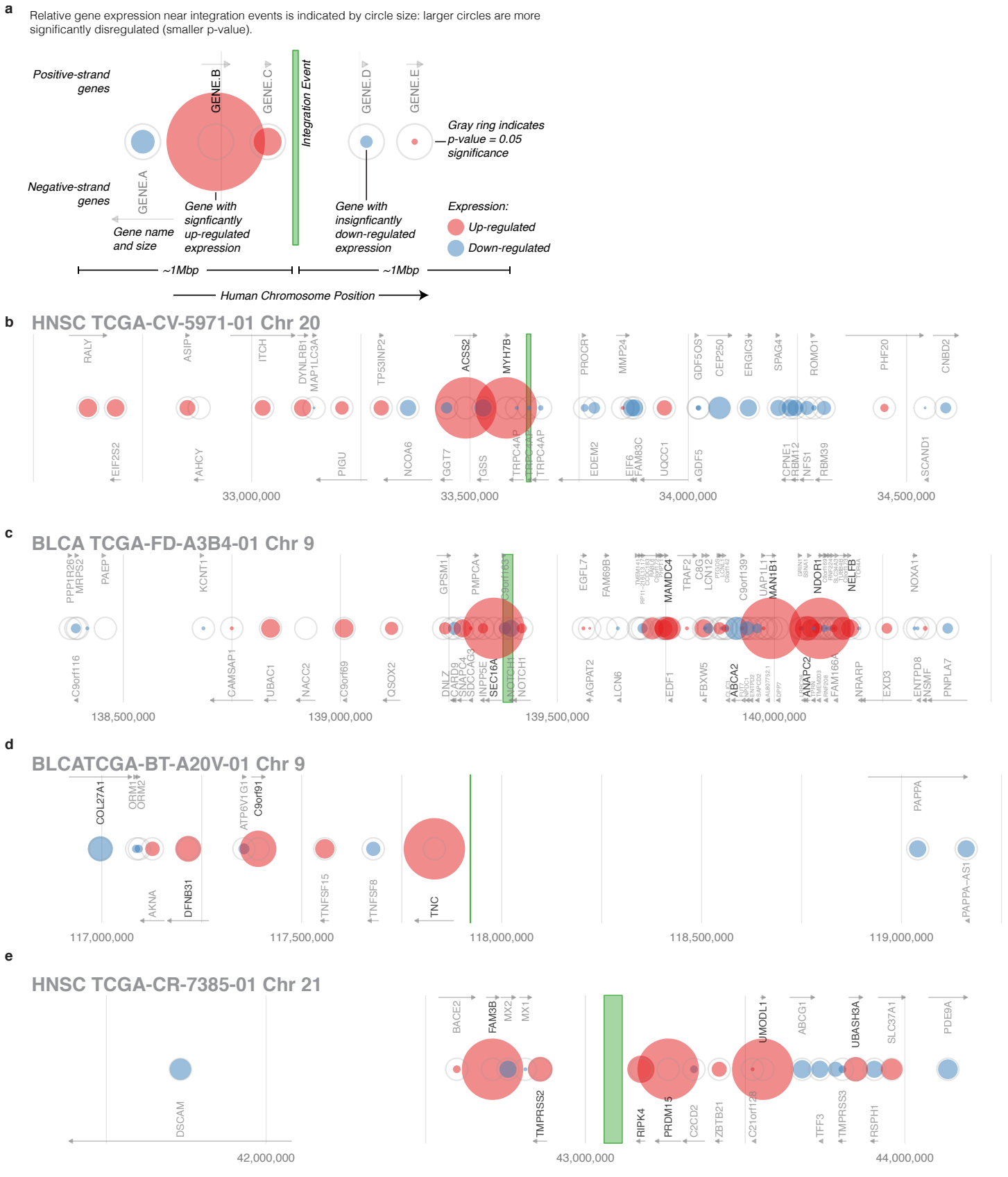
Project Data

- Clear separation of input, output data
- Can be fully reproduced with Workflow Script
- Retains fingerprint of Workflow Scripts and Core Apps used to generate it

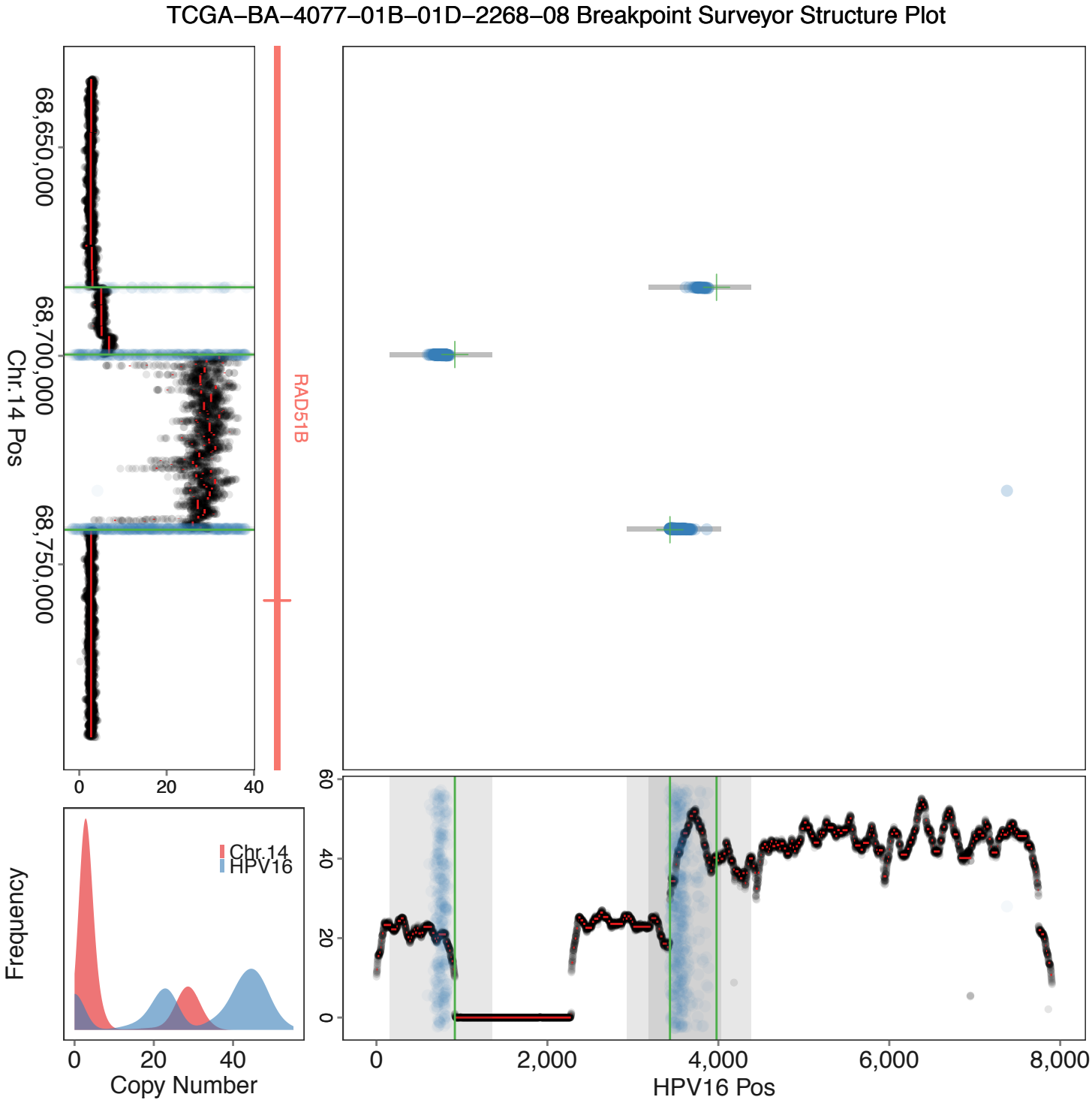
Supplementary Figure 1. Breakpoint Surveyor workflow and software architecture. a) Aligned WGS data are downloaded from a sequencing center, re-aligned to a human + virus reference in the case of virus integration analyses, and processed using a variety of SV-detection tools to obtain breakpoint positions (BPC file). Such breakpoints are clustered to create the PlotList, which defines target regions for further analysis and visualization. (Optional) To evaluate gene expression in such regions, RNA-Seq data for both case and control samples are downloaded and processed to obtain expression values (RPKM) for exons in each region of interest. By comparing case and control, the degree of expression dysregulation can be quantified as a P-value and visualized. Read depth in regions of interest is evaluated, while preliminary breakpoint predictions may be refined by contig assembly. Each analysis is drawn as a separate panel, saved in an intermediate format (GGP), and assembled as panels in a final plot, creating one figure (PDF) per region of interest. b) BPS consists of three hierarchical layers, each typically separately version controlled, to support software reuse, scalability, and data tracking. Core BPS apps are task-specific, locale-independent programs. These are invoked by workflow scripts which process lists of datasets, call various analysis programs, and implement locale-specific details like clustering. These scripts process and generate project data.



Supplementary Figure 2. Structure plots of integration events based on tumor WGS data for additional TCGA samples. a) (i) Breakpoint panel displays breakpoints with virus, human positions as coordinates. (ii, iii) Groups of breakpoints co-localized with copy number (CN) changes define an integration event. (iv) CN changes are illustrated with CN histogram, e.g., two distinct peaks in virus CN suggest two or more virus populations. b) Interpretation of representative integration event. (i) Schematic of simple HPV integration event illustrates breakpoints aligned with CN changes in both human and virus genomes. (ii) Such CN changes suggest duplication of human segment and truncation in circular HPV genome. (iii) Human-virus breakpoints mark genomic junctions and suggest a chimeric episomal structure. (iv) Experimental evidence (*Parfenov et al.*, 2014) supports presence of extrachromosomal human-virus episomes in tumor. Actual integration events are typically more complex, as shown in panels c-f. c) Two breakpoints mark boundaries of HPV16 integration event on chromosome 20, with CN increased five-fold within integration event. Integration events were also observed on chromosomes 5 and 21 in this sample. Histogram shows two peaks in virus CN, suggesting distinct viral populations. d) CN decreases in integration event flanked by breakpoints on chromosome 9, and HPV56 virus is truncated. e) Integration event consists of two breakpoints flanking a truncation in HPV45 genome, which are 49bp apart along chromosome 9, with no discernable CN changes. f) Complex integration event on chromosome 21 is marked by ten distinct human-virus breakpoints and many human-human breakpoints. Complex CN changes co-occur with breakpoints for both human and HPV16 genomes and are reflected in CN histogram.

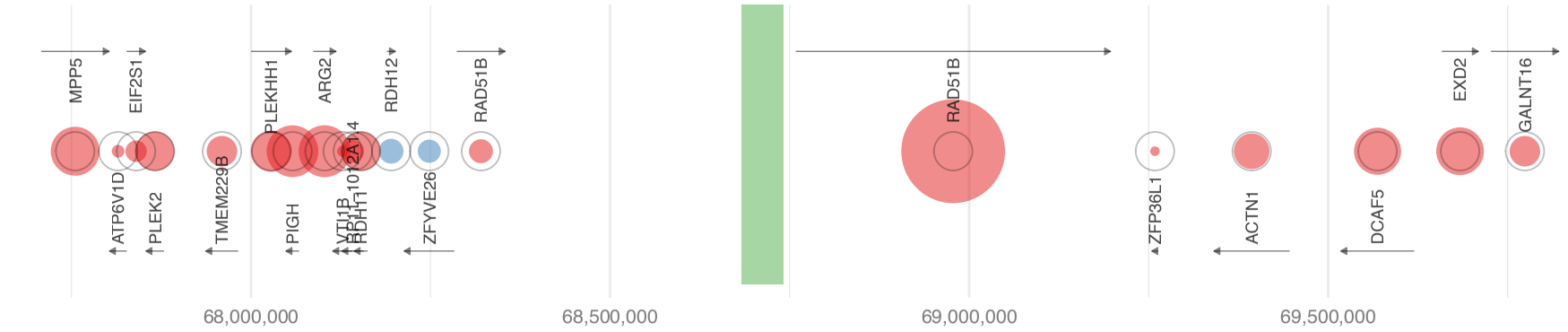


Supplementary Figure 3. Gene expression plots in the vicinity of integration events for additional TCGA samples in Supplementary Figure 2. a) Key for panels b-e. Relative gene expression near integration events is indicated by circle size, with large red and blue circles indicating significant up- and down-regulation, respectively. Here, we provisionally choose p-value = 0.05 as a significance threshold. b) Gene *TRPC4AP* spans integration event in sample TCGA-CV-5971-01. Exon expression upstream, downstream, or within the integration event is not significantly dysregulated, but the upstream gene *ACGS2* is found to be significantly upregulated. c) Genes *C9orf163* and *NOTCH1* intersect integration event and have insignificantly downregulated expression, while expression of downstream genes *SEC16A*, *MAN1B1*, and *NDOR1* is significantly increased. d) Gene *TNC* downstream of integration event as well as upstream gene *C0orf91* both have significantly upregulated expression. e) Genes upstream as well as downstream of the integration event are upregulated, including *FAM3B, PRDM15, UMODL1, RIPK4*, and *TMPRSS2*.

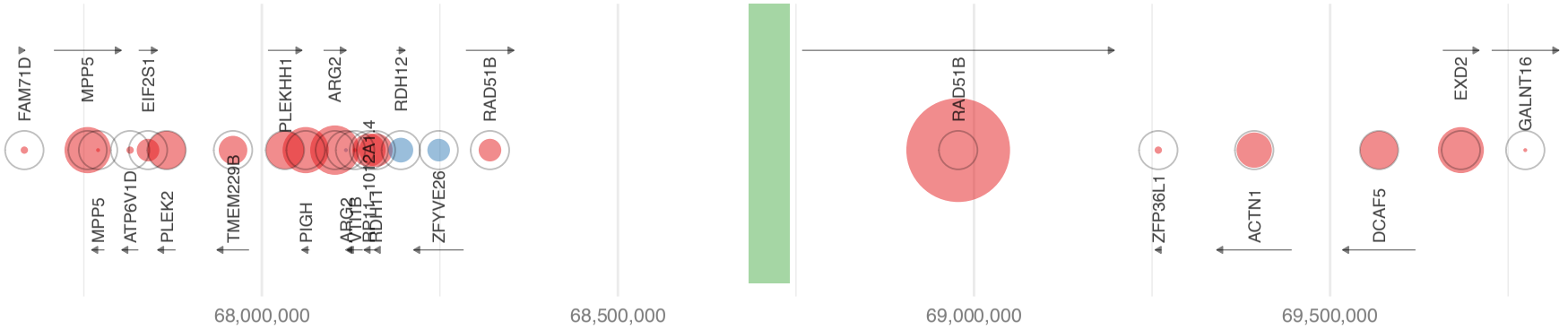


Supplementary Figure 4. Structure plot for TCGA-BA-4077-01, TCGA_virus workflow. This figure is equivalent to Figure 1a in manuscript.

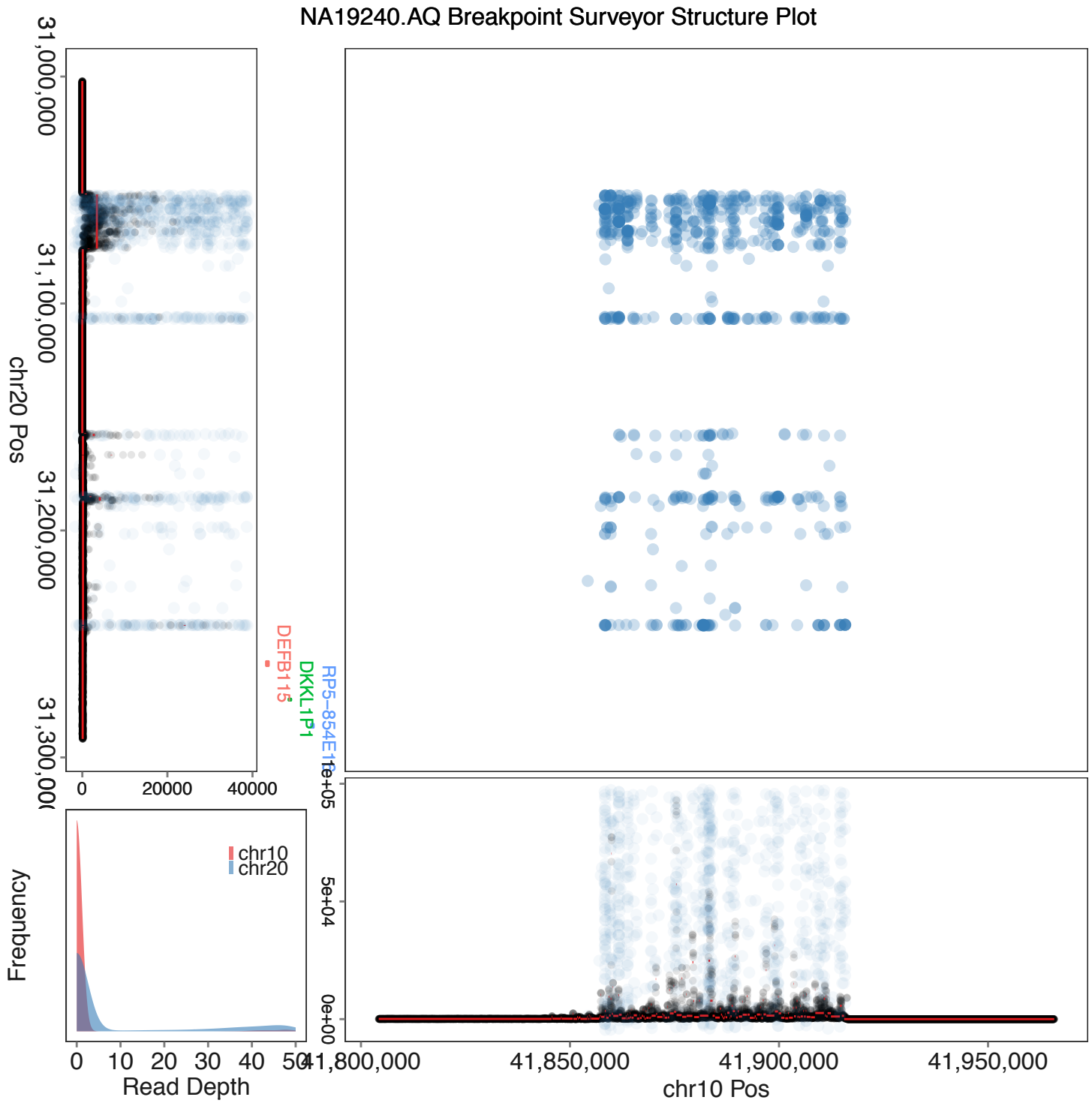
a. TCGA-BA-4077-01B-01D-2268-08 RPKM Expression Plot



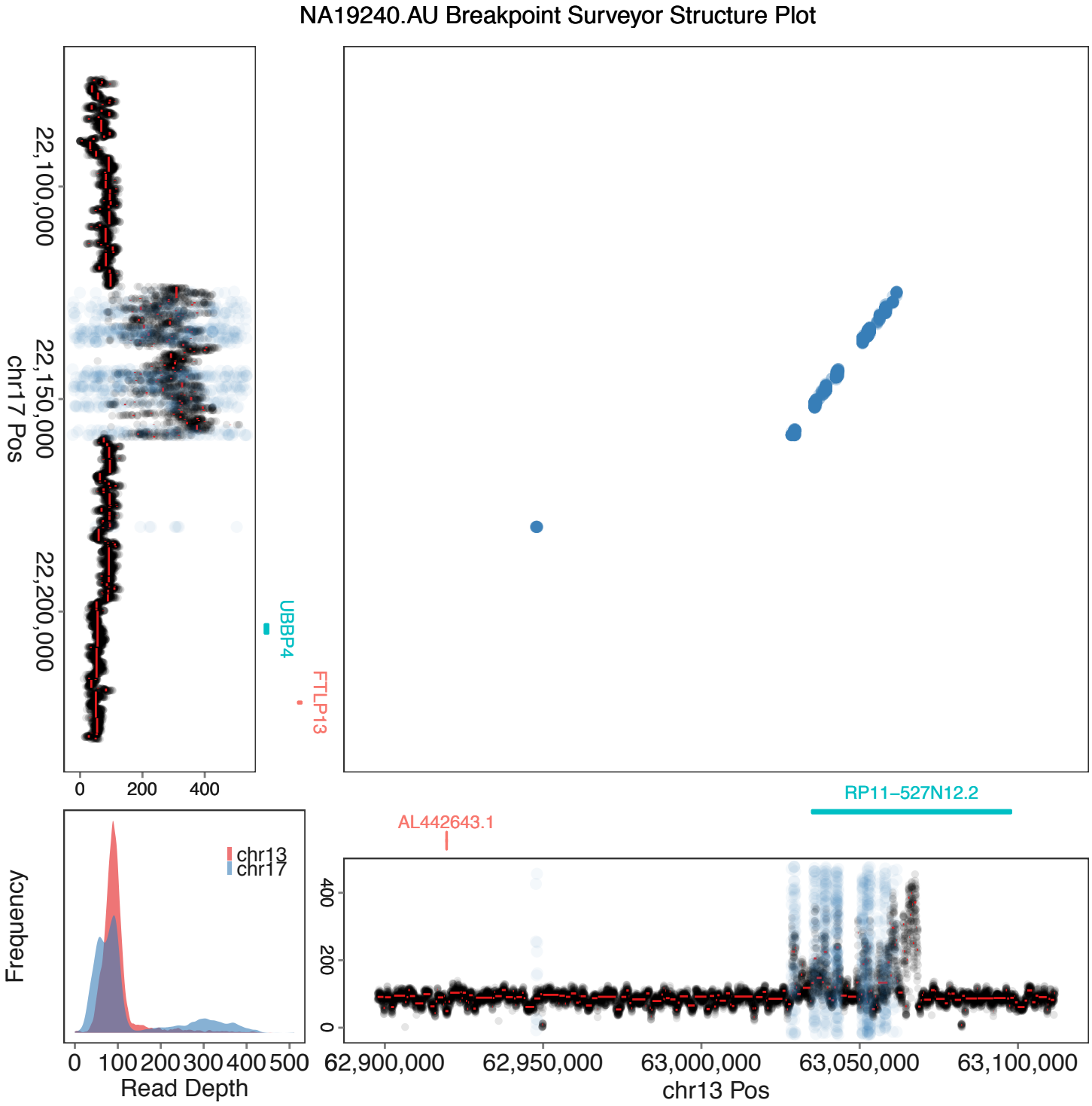
b. TCGA-BA-4077-01B-01D-2268-08 RSEM Expression Plot



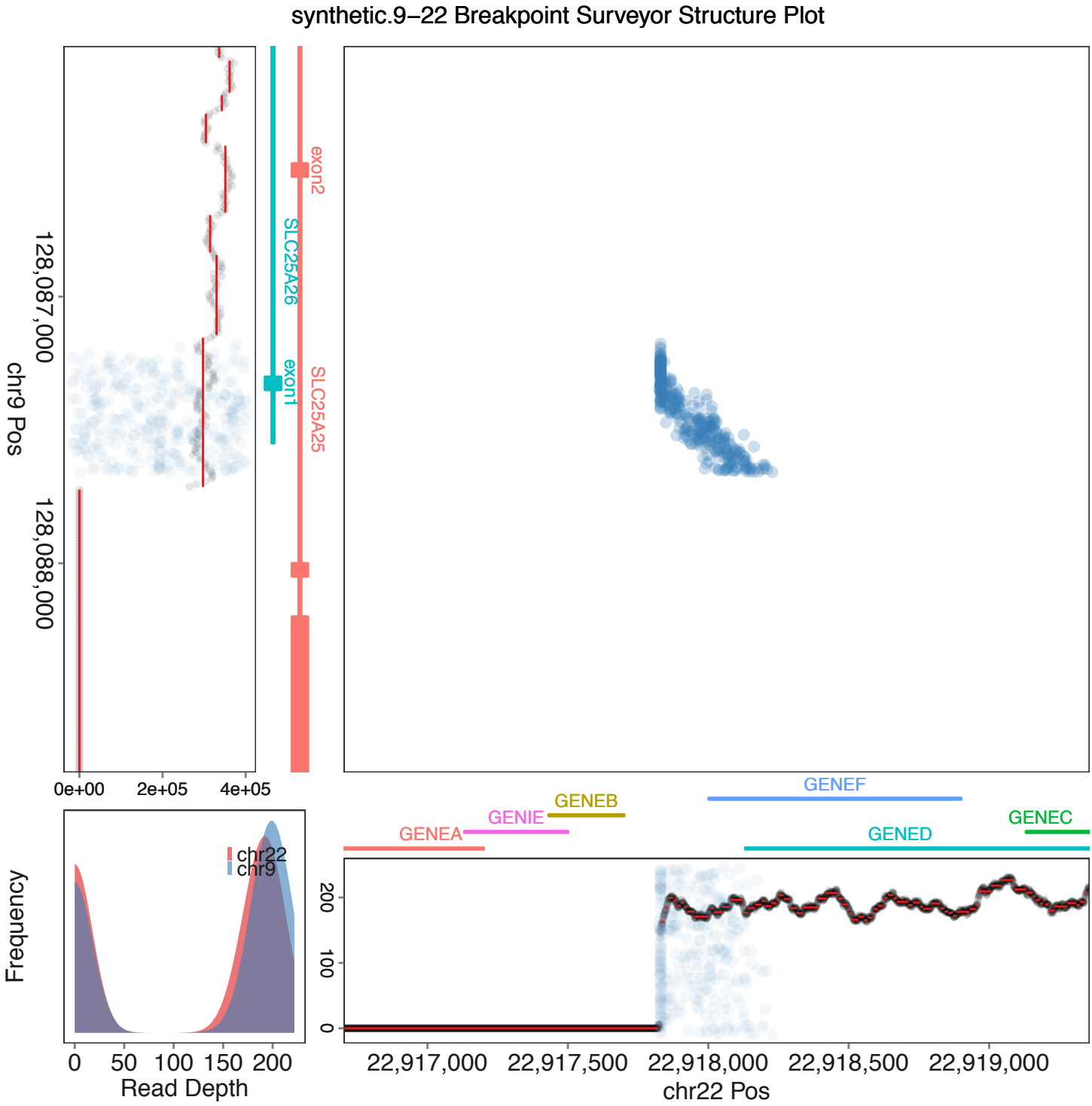
Supplementary Figure 5. Expression plots for TCGA-BA-4077-01 in TCGA Virus workflow. Panel a illustrates gene expression based on analysis of RNA-Seq data and is equivalent to Figure 1b in the manuscript. Panel b illustrates gene expression based on an alternative analysis of preprocessed TCGA RPKM data. The two workflows yield expression figures which are similar.



Supplementary Figure 6. Structure plot for event “AQ” in 1000SV workflow. The 1000SV workflow investigates interchromosomal human-human breakpoints in a publicly available human sample from the 1000 Genomes project, NA19240, which was sequenced at high (80X) coverage. AQ event is between chr10 and chr20, and has a characteristic signature of no correlation between discordant read positions on the two chromosomes, together with a a spike in copy number. This event likely represents anomalous mapping between two repetitive regions.



Supplementary Figure 7. Structure plot for event “AU” in 1000SV workflow. The 1000SV workflow investigates interchromosomal human-human breakpoints in a publicly available human sample from the 1000 Genomes project, NA19240, which was sequenced at high (80X) coverage. AU between chr13 and chr17 has a discordant read pattern distinct from AQ (Supplementary Figure 6), with the reads falling cleanly on a diagonal. This event is likely a tandem duplication.



Supplementary Figure 8. Structure plot for synthetic breakpoint in Synthetic workflow. The Synthetic branch generates an interchromosomal translocation by concatenating two reference segments, then creating synthetic (simulated) reads in this region. Additional gene and exon annotation functionality, including exon labels, is demonstrated with synthetic gene and exon definitions.