Supplementary Information

A robust benchmark for detection of germline large deletions and insertions

Supplementary Note 1: Process to form SV Benchmark

Benchmark Callset Integration Process

1. **Discovery: 68 input variant call sets**
2. **Compare SVs:** Sequence-resolved variants with at least 20% sequence similarity were clustered using the SVanalyzer merge command (https://github.com/nhansen/SVanalyzer). SVanalyzer merges variants by aligning and comparing their extended alternate haplotypes rather than by using size and overlap rules. Pairs of variants whose alternate haplotypes have normalized edit distance and size difference less than or equal to 20% of the length of the extended haplotype region are considered to be matches and clustered into single variants. See the section "Clustering of sequence-resolved variants with SVmerge" for a more detailed description.
3. Discovery Support: Variants supported by at least two technologies (counting BioNano and Nabsys) or by at least 5 callsets from a single technology were processed further.
4. Evaluate/Genotype: Variants from #3 were genotyped using svviz2 with the four datasets in Table 1. Genotypes from Illumina and 10x were excluded in tandem repeats >100 bp in length, and genotypes from PacBio were excluded in tandem repeats >10000 bp. Genotypes from all datasets were excluded in segmental duplications >10000 bp. If the genotypes from all remaining datasets were concordant, and PacBio supported a genotype of heterozygous or homozygous variant, then the variant was included in downstream analyses.
5. Filter Complex: If two or more supported variants ≥50 bp were within 1000 bp of each other, they were excluded because they are potentially complex or inaccurate.

Benchmark Regions Integration Process

1. Find Diploid Assembled Regions: Using 3 PacBio-based (MsPAC, Phased-SV, and Falcon-unzip) and 1 10x-based (supernova) assemblies (results and methods at ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/NHGRI_SVrefine_04122018/), find regions for each assembly that are covered by exactly one contig for each haplotype using assembly bedgraphs generated by SVRefine.
2. Combine Regions: Find the union of the regions from the assemblies in #1.
3. Discover Variants: Use SVRefine to call variants from the 3 assemblies (vcfs at ftp link in #1)
4. Compare Variants: Use SVRefine to compare variants from each assembly to our v0.6 PASS calls for HG002, allowing them to be up to 20 % different in all 3 distance measures, and only keep variants not matching a v0.6 call.
   a. Cluster the remaining variants from all assemblies and keep any that are supported by at least one long read assembly
5. Trim Regions: From the regions defined in #2, remove regions around unresolved calls, including
   a. regions around variants remaining after #4, using svwiden's repeat-expanded coordinates, and expanded further to include any overlapping repetitive regions from Tandem Repeat Finder, RepeatMasker SimpleRepeats, and RepeatMasker LowComplexity, plus 50 bp on each end.

b. regions in the Tier 2 bed file of unresolved and clusters of variants, unless the Tier 2 region overlaps a Tier 1 PASS call.

c. The HG002_SVs_Tier1_noVDJorXorY_v0.6.2.bed further excludes:

   i. Regions that undergo somatic V(D)J recombination in the immunoglobulin and T cell receptor loci (hg19.vdjwithTCR.bed)

   ii. Regions with N's in the reference plus 15 kb on either side, to minimize complications in benchmarking near gaps in the reference

   iii. Regions on the X and Y chromosomes, since our benchmark is designed for diploid regions

Supplementary Table 1: Heuristics for genotype determination from each dataset. Svviz was used to determine the genotype for HG002 using different heuristics for each dataset. The cut-offs for weighted alternate (ALT) and reference (REF) counts were determined manually from looking at distributions for different size ranges. For each technology, genotypes were determined based on the proportion of reads supporting the SV. If the ALT and REF counts did not meet the criteria in this table for a particular dataset, the genotype was considered uncertain. In addition, the genotype was considered uncertain for PacBio if it overlapped a tandem repeat longer than 10 kbp, for Illumina and 10x Genomics if it overlapped a tandem repeat longer than 100 bp, and for all datasets if it overlapped a segmental duplication >10 kbp.

Dataset(s)	**Minimum Coverage**	**Proportion of reads supporting SV (x = ALT/(REF+ALT))**	**Genotype Label**
PacBio	$ALT+REF\geq 8$	$x<0.1$	0/0
		$0.25<x<0.75$	0/1
		$x>0.9$	1/1
Illumina 250bp and Illumina Mate-Pair	$ALT+REF\geq 8$	$x<0.05$	0/0
		$0.1<x<0.9$	0/1
		$x>0.95$	1/1
Haplotype-partitioned 10x Genomics and PacBio (haplotype in subscript)	$ALT_1+REF_1\geq 5$ AND $ALT_2+REF_2\geq 5$	$x_1<0.05$ AND $x_2<0.05$	0/0
		($x_1>0.95$ AND $x_2<0.05$) OR ($x_2>0.95$ AND $x_1<0.05$)	0/1
		$x_1>0.95$ AND $x_2>0.95$	1/1

Tier 2 Benchmark Integration Process

We designed the draft Tier 2 benchmark set as a less conservative set of regions in which there appeared to be good evidence for at least one SV, but there were multiple SVs within 1 kb, multiple SVs within a tandem repeat, and/or different SV callers had different results for reasons that were not yet resolved. The process for forming the Tier 2 regions was:

1. **Add 1000 bp to each side of any variants with the FILTER "ClusteredCalls" or "MaxEditDistgt04" and merge regions separated by <50 bp**. Expand these regions to completely encompass any overlapping tandem repeats (after merging tandem repeats within 50 bp and adding 5 bp on each side, available at https://github.com/jzook/genome-data-integration/blob/master/StructuralVariants/NISTv0.6/repeats_trfSimplerepLowcomplex_merged50_slop5.bed.gz)
2. **Take any variants ≥50 bp with the FILTER "NoConsensusGT"**, expand these regions to completely encompass any overlapping tandem repeats (after merging tandem repeats within 50bp and adding 5 bp on each side), merge regions separated by <50 bp, and add 50 bp to each side.
3. **After removing variants discovered by at least 2 technologies or 5 callsets (the inverse of those tested in the Tier 1 process above), cluster variants within 1000 bp (without considering type or sequence change), and find regions with clusters having calls from at least 2 technologies or 5 callsets.** Expand these regions to completely encompass any overlapping tandem repeats (after merging tandem repeats within 50bp and adding 5 bp on each side).
4. **Remove any regions from #1 and #2 that have any overlap with a Tier 1 benchmark call, and take the union of the resulting regions and the regions from** #3. Merge regions within 50bp, and the result is the Tier 2 bed.

Clustering of sequence-resolved variants with SVmerge

Structural variants are frequently flanked by stretches of repeated sequence which obscure the true position of the structural event. For this reason, we used a repeat-aware method to compare sequence-resolved structural variants, rather than relying on size and overlap-based rules. The program SVmerge, part of the SVanalyzer package (http://github.com/nhansen/SVanalyzer) was used to compare pairs of non-identical structural variant calls and cluster them based on distance measures. To calculate these measures of distance, SVmerge constructs alternate haplotype sequences corresponding to a common, widened region of the reference which includes all bases altered by either of the two variants. The two resulting alternate haplotypes are then compared by global alignment (Needleman Wunsch, as implemented in the edlib software library),[50] and the resulting alignment is used to calculate three normalized measures of difference: (1) the edit distance between the two alternate haplotypes, (2) the size difference in base pairs between the two alternate haplotypes, and (3) the maximum shift of coordinates in the global alignment between the two haplotypes. Each of these distances is then normalized by dividing by the mean length of the longer allele (reference or alternate) for each of the two variants.

To combine the structural variant calls into clusters, SVmerge creates an undirected graph in which variant calls are nodes and edges exist between pairs of calls having all three distances less than or equal to specified maximum values. Variants are then merged into a single cluster if they are within the same connected component of the resulting graph. If any variants match exactly, one of the exactly matching variants is selected for output in the REF and ALT fields as the representative call for the cluster. If no variants match exactly, a representative call is randomly selected from the cluster.

Trio+linked read phased vcf and haplotype-partitioned bam files

To produce a chromosome-length phasing of small variants for the Ashkenazim trio, we combined variant calls from Real Time Genomics (ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/Rutgers_IlluminaHiSeq300X_rtg_11052015/rtg_allCallsV2.vcf.gz) with phased blocks produced by 10x Genomics in the following vcfs:

ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/10XGenomics_ChromiumGenome_LongRanger2.1_09302016/NA24143_hg19/NA24143_hg19_phased_variants.vcf.gz

ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/10XGenomics_ChromiumGenome_LongRanger2.1_09302016/NA24149_hg19/NA24149_hg19_phased_variants.vcf.gz

ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/10XGenomics_ChromiumGenome_LongRanger2.1_09302016/NA24385_hg19/NA24385_hg19_phased_variants.vcf.gz

The single sample 10x Genomics VCF files were combined into multi-sample VCF using bcftools and all VCFs were split by chromosome (to facilitate easy parallelization with Snakemake). Then, WhatsHap (version 0.15+14.ga105b78)[51] was used in pedigree-aware mode[52] using the following command line:

```
whatshap phase --ped AJ.ped --indels --reference hg19.fasta rtg.vcf 10x-merged.vcf | bgzip > output.vcf
```

This vcf with whatshap haplotag was used to partition reads in the PacBio bam files for svviz and manual curation.

Supplementary Note 2: SV Discovery Callsets for Integration

Illumina-based SV Discovery Callsets

Cortex

This callset, generated jointly for the trio, used cortex[53] (version 1.0.5.21, code at http://cortexassembler.sourceforge.net/index_cortex_var.html) with default parameters and Illumina HiSeq 300x 2x150 bp data for the AJ trio. Only variants with "PASS" status from the raw callset were included. Sites with Mendelian inconsistencies were identified and removed (47048 sites). Sites with mislabeling were also corrected (526 sites). Total variant count was 3130512, including 2780684 SNPs, 164402 deletions, 150560 insertions, 29 INV_INDEL, and 34837 COMPLEX variants, and the output VCF is under:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/NCBI_IlluminaHiSeq300X_cortex_09042015/.
The input fastqs are under:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG002_NA24385_son/NIST_HiSeq_HG002_Homogeneity-10953946/HG002_HiSeq300x_fastq/
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG003_NA24149_father/NIST_HiSeq_HG003_Homogeneity-12389378/HG003_HiSeq300x_fastq/
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG004_NA24143_mother/NIST_HiSeq_HG004_Homogeneity-14572558/HG004_HiSeq300x_fastq/

Manta

These callsets, generated independently for each individual in the trio, used manta[54] (version 0.27.1, code at https://github.com/Illumina/manta) with default parameters and Illumina HiSeq 30x (downsampled by read group) 2x150bp data for the AJ trio mapped by BWA MEM v1.5.0 against the hs37d5 reference genome. The output VCFs are at:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/DNAnexus_AndrewC_Illumina_Callers_Sep2016/HG002/HG002.140528_D00360_0018_AH8VC6ADXX.realigned.recalibrated.manta.diploidSV.vcf
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/DNAnexus_AndrewC_Illumina_Callers_Sep2016/HG003/HG003.140701_D00360_0032_AHA0KGADXX.realigned.recalibrated.manta.diploidSV.vcf
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/DNAnexus_AndrewC_Illumina_Callers_Sep2016/HG004/HG004.140818_D00360_0046_AHA5R5ADXX.realigned.recalibrated.manta.diploidSV.vcf
The input fastqs, each downsampled to 30x, are under:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG002_NA24385_son/NIST_HiSeq_HG002_Homogeneity-10953946/HG002_HiSeq300x_fastq/
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG003_NA24149_father/NIST_HiSeq_HG003_Homogeneity-12389378/HG003_HiSeq300x_fastq/

ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG004_NA24143_mother/NIST_HiSeq_HG004_Homogeneity-14572558/HG004_HiSeq300x_fastq/

GATK HaplotypeCaller

These callsets, generated independently for each individual in the trio, used GATK HaplotypeCaller[55] (version 3.5, code at https://hub.docker.com/r/broadinstitute/gatk3) with high-sensitivity settings from Illumina HiSeq 300x 2x150 bp data for the AJ trio. Specifically, special options were '-stand_call_conf 2 -stand_emit_conf 2 -A BaseQualityRankSumTest -A ClippingRankSumTest -A Coverage -A FisherStrand -A LowMQ -A RMSMappingQuality -A ReadPosRankSumTest -A StrandOddsRatio -A HomopolymerRun -A TandemRepeatAnnotator'. The gVCF output was converted to variant call format (VCF) using GATK Genotype gVCFs for each sample independently. The output VCFs were filtered to calls >19bp in size and are at:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/release/AshkenazimTrio/HG002_NA24385_son/NISTv3.3.2/GRCh37/supplementaryFiles/inputvcfsandbeds/HG002_GRCh37_CHROM1-MT_novoalign_Ilmn150bp300X_GATKHC.vcf.gz
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/release/AshkenazimTrio/HG003_NA24149_father/NISTv3.3.2/GRCh37/supplementaryFiles/inputvcfsandbeds/HG003_GRCh37_CHROM1-Y_novoalign_Ilmn150bp300X_GATKHC.vcf.gz
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/release/AshkenazimTrio/HG004_NA24143_mother/NISTv3.3.2/GRCh37/supplementaryFiles/inputvcfsandbeds/HG004_GRCh37_CHROM1-MT_novoalign_Ilmn150bp300X_GATKHC.vcf.gz
The input bam files are under:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG002_NA24385_son/NIST_HiSeq_HG002_Homogeneity-10953946/NHGRI_Illumina300X_AJtrio_novoalign_bams/
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG003_NA24149_father/NIST_HiSeq_HG003_Homogeneity-12389378/NHGRI_Illumina300X_AJtrio_novoalign_bams/
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG004_NA24143_mother/NIST_HiSeq_HG004_Homogeneity-14572558/NHGRI_Illumina300X_AJtrio_novoalign_bams/

Freebayes

These callsets, generated independently for each individual in the trio, used freebayes[56] (version 0.9.20, code at https://github.com/ekg/freebayes) with high-sensitivity settings from Illumina HiSeq 300x 2x150bp data for the AJ trio. Specifically, special options were '-F 0.05 -m 0 --genotype-qualities'. The output VCFs were filtered to calls >19bp in size and are at:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/release/AshkenazimTrio/HG002_NA24385_son/NISTv3.3.2/GRCh37/supplementaryFiles/inputvcfsandbeds/HG002_GRCh37_CHROM1-MT_novoalign_Ilmn150bp300X_FB.vcf
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/release/AshkenazimTrio/HG003_NA24149_father/NISTv3.3.2/GRCh37/supplementaryFiles/inputvcfsandbeds/HG003_GRCh37_CHROM1-Y_novoalign_Ilmn150bp300X_FB.vcf.gz

ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/release/AshkenazimTrio/HG004_NA24143_mother/NISTv3.3.2/GRCh37/supplementaryFiles/inputvcfsandbeds/HG004_GRCh37_CHROM1-MT_novoalign_Ilmn150bp300X_FB.vcf.gz
The input bam files are under:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG002_NA24385_son/NIST_HiSeq_HG002_Homogeneity-10953946/NHGRI_Illumina300X_AJtrio_novoalign_bams/
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG003_NA24149_father/NIST_HiSeq_HG003_Homogeneity-12389378/NHGRI_Illumina300X_AJtrio_novoalign_bams/
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG004_NA24143_mother/NIST_HiSeq_HG004_Homogeneity-14572558/NHGRI_Illumina300X_AJtrio_novoalign_bams/

FermiKit for 150 bp and 250 bp Illumina datasets

These callsets, generated independently for each individual in the trio, used fermikit[57] (version 6fc8bbb3, code at https://github.com/lh3/fermikit in precisionFDA app at https://precision.fda.gov/apps/app-BvJPP100469368x7QvJkKG9Y-1) with default settings from Illumina HiSeq 50x (downsampled to two flow cells) 2x150 bp data and from Illumina HiSeq 45x 2x250 bp data for the AJ trio. Specifically, commands were '/fermi.kit/fermi2.pl unitig -s3g -t$(nproc) -l$(read length) -p genome reads.fq.gz > genome.mak', 'make -f genome.mak', and 'fermi.kit/run-calling -t$(nproc) bwa-indexed-ref.fa genome.mag.gz | sh'. The output VCFs were filtered to calls >19bp in size and are under:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/DNAnexus_fermikit_160505/

The input fastqs are under:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG002_NA24385_son/NIST_HiSeq_HG002_Homogeneity-10953946/HG002_HiSeq300x_fastq/
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG003_NA24149_father/NIST_HiSeq_HG003_Homogeneity-12389378/HG003_HiSeq300x_fastq/
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG004_NA24143_mother/NIST_HiSeq_HG004_Homogeneity-14572558/HG004_HiSeq300x_fastq/

MetaSV

These callsets, generated independently for each individual in the trio, used MetaSV[58] (version 0.5, code at https://github.com/bioinform/metasv) with default settings from Illumina HiSeq 60x 2x150 bp data for the AJ trio. Specifically, special options were '--boost_sc --filter_gaps --keep_standard_contigs --isize_mean 550 --isize_sd 145 --svs_to_assemble INS INV DEL DUP --svs_to_softclip INS INV DEL DUP --svs_to_report INS INV DEL DUP --max_ins_cov_frac 2 --min_support_frac_ins 0.015 --min_support_ins 15 --max_ins_intervals 24000 --mean_read_length 150 --mean_read_coverage 60 --age_window 50'. Results from BreakSeq, BreakDancer, Pindel, and CNVnator were used as inputs into MetaSV. The output VCFs were filtered to PASS calls and are under:

ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/BINA_Roche_MetaSV_10142016
The input bam files are under:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG002_NA24385_son/NIST_HiSeq_HG002_Homogeneity-10953946/NHGRI_Illumina300X_AJtrio_novoalign_bams/
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG003_NA24149_father/NIST_HiSeq_HG003_Homogeneity-12389378/NHGRI_Illumina300X_AJtrio_novoalign_bams/
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG004_NA24143_mother/NIST_HiSeq_HG004_Homogeneity-14572558/NHGRI_Illumina300X_AJtrio_novoalign_bams/

TNscope

These callsets, generated for HG002 only, used TNscope[59] (version 201704, from https://www.sentieon.com/) with default settings from Illumina HiSeq 300x 2x150 bp data for the AJ son. Filters removed sites with a total depth of greater than or equal to 230 (calculated as the sum of the sample AD), QUAL less than or equal to 52, or a faction of reads support the alternate allele less than 0.03. A script was used to convert the breakpoints produced by TNscope to a sequence-resolved ref/alt format for integration with the NIST callsets. The script used the orientation and size of the breakpoint to classify the breakpoint as either a deletion, duplication or inversion. The output VCFs were filtered to calls >19bp in size and are at:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/Sentieon_tnscope_05052017/HG002_300x_tnscope_hq_altallele_head.vcf.gz
The input fastqs are under:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG002_NA24385_son/NIST_HiSeq_HG002_Homogeneity-10953946/HG002_HiSeq300x_fastq/

Scalpel

These callsets, generated independently for each individual in the trio, used scalpel[60] (version 0.4.1 beta, code at http://scalpel.sourceforge.net/) with default settings and window size 600 from Illumina HiSeq 300x 2x150 bp data for the AJ trio. The output VCFs were filtered to calls >19 bp in size and are under:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/BU_HiSeq300x_scalpel_v0.4.1_04202017/
The input bam files are under:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG002_NA24385_son/NIST_HiSeq_HG002_Homogeneity-10953946/NHGRI_Illumina300X_AJtrio_novoalign_bams/
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG003_NA24149_father/NIST_HiSeq_HG003_Homogeneity-12389378/NHGRI_Illumina300X_AJtrio_novoalign_bams/
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG004_NA24143_mother/NIST_HiSeq_HG004_Homogeneity-14572558/NHGRI_Illumina300X_AJtrio_novoalign_bams/

SvABA

These callsets, generated jointly for the trio, used SvABA[35] (version 0.2.1, code at https://github.com/walaj/svaba) with default settings from Illumina HiSeq 300x 2x150 bp data for the AJ trio. SvABA de-novo indel and SV calls were made with the proband BAM as the primary BAM and the parent BAMs as controls (-t <proband> -n <maternal> -n <paternal>). dbSNP v138 was used as input to the -D flag to increase confidence that de novo variants were not false negative variants from the controls. SvABA calls are produced using the breakend (BND) format, and were converted to DEL format by selecting those SVs with a +, - orientation pair, indicating a likely deletion. The variants are captured in both an SV VCF using the BND format for larger SVs, and an indel VCF for smaller SVs (< ≅100 bp). The output VCFs were filtered to calls >19 bp in size and are under:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/Broad_svaba_05052017/
The input bam files are under:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG002_NA24385_son/NIST_HiSeq_HG002_Homogeneity-10953946/NHGRI_Illumina300X_AJtrio_novoalign_bams/
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG003_NA24149_father/NIST_HiSeq_HG003_Homogeneity-12389378/NHGRI_Illumina300X_AJtrio_novoalign_bams/
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG004_NA24143_mother/NIST_HiSeq_HG004_Homogeneity-14572558/NHGRI_Illumina300X_AJtrio_novoalign_bams/

Krunch

These callsets, generated independently for each individual in the trio used a method under development called Krunch (code at https://github.com/hansenlo/SeqDiff) from Illumina HiSeq 300x 2x150 bp data for the AJ trio. Krunch is a method developed to call variants that allows for direct comparison of sequence libraries with a reference genome or to each other without requiring the alignment of reads to a reference genome. The method is based on comparative indexing of DNA kmers unique to one read library compared to the other or to the reference genome. This identifies reads that share the same variant because they share the same unique kmers(s). We then assemble reads containing the same set of unique words into a contig, and align the contig or edges of the contig to the reference genome, allowing us to accurately call the variant type and position. This approach detects single nucleotide polymorphism (SNPs), small indels, medium and large structural variants, both germline and somatic. The output VCFs were filtered to calls >19 bp in size and are under:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/Stanford_Krunch_05052017/
The input fastqs are under:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG002_NA24385_son/NIST_HiSeq_HG002_Homogeneity-10953946/HG002_HiSeq300x_fastq/
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG003_NA24149_father/NIST_HiSeq_HG003_Homogeneity-12389378/HG003_HiSeq300x_fastq/
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG004_NA24143_mother/NIST_HiSeq_HG004_Homogeneity-14572558/HG004_HiSeq300x_fastq/

Spiral Genetics Anchored Assembly

These callsets, generated independently for each individual in the trio, used the Spiral Genetics Anchored Assembly variant caller (version May 2015, from https://www.spiralgenetics.com/) with high-sensitivity settings from Illumina HiSeq 50x (downsampled by run) 2x150bp data for the AJ trio. Specifically, the commands were 'spiral kmerize $sample ${sample}kmers ${sample}kmer_quality_report', 'spiral correct_reads $sample ${sample}kmers ${sample}corrected_reads --min-kmer-score 8', and 'spiral find_variants ${sample}corrected_reads hg19 ${sample}variants'. The output VCFs were filtered to sequence-resolved (not breakend/BND) calls >19bp in size and are under:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/NIST_Stanford_HiSeq300x_SpiralGenetics_vcf_06042015/
The input fastqs (only run 6 for HG002 and HG004 and only run 3 for HG003) are under:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG002_NA24385_son/NIST_HiSeq_HG002_Homogeneity-10953946/HG002_HiSeq300x_fastq/
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG003_NA24149_father/NIST_HiSeq_HG003_Homogeneity-12389378/HG003_HiSeq300x_fastq/
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG004_NA24143_mother/NIST_HiSeq_HG004_Homogeneity-14572558/HG004_HiSeq300x_fastq/

Spiral Genetics BioGraph Refinement

This callset, generated only for HG002, used the Spiral Genetics BioGraph variant caller (version 1.1, from https://www.spiralgenetics.com/) taking in the union of all variant calls >19bp generated prior to November 11, 2016. The output VCF is under:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/Spiral_AJTrio_v1.1_01312017/
The following process was used to (1) Unite a set of GIAB Variants and unite evaluate calls using Spiral force calling. (2) Create a useful database that links all relevant information and produce metrics summarizing the results. All steps in the procedure are coded in Workflow.py. Detailed information is at the end of this Supplementary Note 2.

Seven Bridges Graph Refinement

This callset, generated independently for each individual in the trio, used the Seven Bridges Graph Aligner[57] and GATK HaplotypeCaller[51] (version 3.5, from https://hub.docker.com/r/broadinstitute/gatk3) taking in the union of all variant calls >19bp generated prior to April 14, 2017. Calls are based on alignments produced by the Seven Bridges Graph aligner, using the NIST Union SV callset 170414 from all 3 members of the trio as the contents of the reference graph. Calls are made by GATK HaplotypeCaller by explicitly forcing calls in a wide region around the putative variant sites. The output VCFs are under:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/SevenBridges_GraphGATKRefine_05052017/

10x Genomics-based SV Discovery Callsets

LongRanger

These callsets, generated independently for each individual in the trio, used LongRanger[12] (version 2.1, code at https://github.com/10XGenomics/longranger) with default parameters and 10x Genomics data for the AJ trio (86x, 36x, and 47x coverage for HG002, HG003, and HG004, respectively). Indels >19bp from *_phased_variants.vcf.gz and large deletions from *_deletions.vcf.gz were converted into sequence-resolved vcf format. Vcf and bam files for each genome are under:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/10XGenomics_ChromiumGenome_LongRanger2.1_09302016/

Complete Genomics-based SV Discovery Callsets

CGATools

These callsets, generated independently for each individual in the trio, used CGATools (version 1.8.0, code at http://cgatools.sourceforge.net) with default parameters and Complete Genomics data for the AJ trio (~100x coverage for each genome). Only indels >19bp from the vcfBeta were used since SV calls were not in sequence-resolved format. vcfBeta files for each genome are at:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/CompleteGenomics_RefMaterial_SmallVariants_CGAtools_08082014/son_NA24385_GS000037263-ASM/ASM/vcfBeta-GS000037263-ASM.vcf.bz2
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/CompleteGenomics_RefMaterial_SmallVariants_CGAtools_08082014/dad_NA24149_GS000037264-ASM/ASM/vcfBeta-GS000037264-ASM.vcf.bz2
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/CompleteGenomics_RefMaterial_SmallVariants_CGAtools_08082014/mom_NA24143_GS000037262-ASM/ASM/vcfBeta-GS000037262-ASM.vcf.bz2
Raw data are under:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG002_NA24385_son/CompleteGenomics_normal_RMDNA/
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG003_NA24149_father/CompleteGenomics_normal_RMDNA/
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG004_NA24143_mother/CompleteGenomics_normal_RMDNA/

PacBio-based SV Discovery Callsets

pbsv

These callsets, generated independently for each individual in the trio, used pbsv (version v0.1-prerelease, code at https://github.com/PacificBiosciences/pbsv) with default parameters and continuous long read PacBio data for the AJ trio aligned with NGM-LR 0.2.4 [15] to the hs37d5 reference (ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/reference/phase2_reference_assembly_sequence/README_human_reference_20110707). Duplicate alignments were marked with "pbsvutil markduplicates"

and alignments were chained with "pbsvutil chain" with default parameters. For HG003 and HG004, at least 2 reads and 20% of reads were required. For HG002, at least 3 reads and 20% of reads were required. Only reads with MAPQ ≥ 20 were considered. VCF files for each genome are under:
ftp://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/AshkenazimTrio/analysis/PacBio_pbsv_05052017/
Fastq files for each genome are under:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG002_NA24385_son/PacBio_MtSinai_NIST/
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG003_NA24149_father/PacBio_MtSinai_NIST/
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG004_NA24143_mother/PacBio_MtSinai_NIST/

Multi-technology-based SV Discovery Callsets

HySA

These callsets, generated independently for each individual in the trio, used HySA[62] (commit ID eee31f6, code at https://bitbucket.org/xianfan/hybridassemblysv/overview) with default parameters and merged Illumina HiSeq 300x 2x150bp data and continuous long read PacBio data for the AJ trio. Post filtering includes only one step: removing all calls with just one Illumina read as the support. Vcf files for each genome are under:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/MDAnderson_HySA_05052017/
Fastq files for each genome are under:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG002_NA24385_son/NIST_HiSeq_HG002_Homogeneity-10953946/HG002_HiSeq300x_fastq/
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG003_NA24149_father/NIST_HiSeq_HG003_Homogeneity-12389378/HG003_HiSeq300x_fastq/
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG004_NA24143_mother/NIST_HiSeq_HG004_Homogeneity-14572558/HG004_HiSeq300x_fastq/
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG002_NA24385_son/PacBio_MtSinai_NIST/
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG003_NA24149_father/PacBio_MtSinai_NIST/
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG004_NA24143_mother/PacBio_MtSinai_NIST/

BreakScan

BreakScan is a kmer-based structural variation discovery method, which models insertion and deletion events in reference sequence with breakpoint junctions observed in NGS reads, then subsequently compiles evidence for those junctions from the sequencing reads generated by multiple platforms such as Illumina, 10X Genomics, and Pacific Biosciences. The candidate structural variants are generated from event models and ranked by their supporting evidence. These callsets, including 2,918 deletions and 2,193 insertions, generated only for HG002, used BreakScan (https://github.com/chunlinxiao/BreakScan) with default parameters and reads from Illumina HiSeq 300x 2x150bp data, 10x Genomics data, and error-corrected continuous long read PacBio data for HG002. Only variants supported by at least two technologies are included. VCF files for insertions and deletions are under:
ftp://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/AshkenazimTrio/analysis/NCBI_BreakScan_12072017_v1.1/
Input fastq files for each technology are under:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG002_NA24385_son/NIST_HiSeq_HG002_Homogeneity-10953946/HG002_HiSeq300x_fastq/,
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/10XGenomics_ChromiumGenome_LongRanger2.1_09302016/
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG002_NA24385_son/PacBio_MtSinai_NIST/

Global de novo assembly-based SV Discovery Callsets

SVrefine

These callsets, generated independently for each individual in the trio, used SVRefine (version 0.2, code at https://github.com/nhansen/SVanalyzer) with default parameters and from a variety of global de novo assemblies from different technologies and assembly methods. For assemblies with both unscaffolded fasta files and fasta files scaffolded with Dovetail, we merged calls from unscaffolded assemblies with their Dovetail-scaffolded counterparts using SVmerge.pl and SVcluster.pl from the SVanalyzer, with the commands 'SVmerge.pl --ref $REF --vcf $UNIONVCF > $DISTFILE', 'gunzip -c $VCF | grep -v '#' | awk '{print $3}' > $IDFILE', and 'SVcluster.pl --ids $IDFILE --dist $DISTFILE --relshift 0 --relsizediff 0 --reldist 0 --vcf $VCF > $CLUSTERFILE'. Specifically, HG2_MHAP_plus_Dovetail.clustered.0.0.0.vcf.gz is a merge of HG2_Dovetail_MHAP.l100c500.vcf.gz and HG2_PBcR_MHAP.l100c500.vcf.gz, HG2_Falcon_plus_Dovetail.clustered.0.0.0.vcf.gz is a merge of HG2_Dovetail_Falcon.l100c500.vcf.gz and HG2_p_and_a_ctg.l100c500.vcf.gz, and nd HG2_DISCOVAR_plus_Dovetail.clustered.0.0.0.vcf.gz is a merge of HG2_Dovetail_DISCOVAR.l100c500.vcf.gz and HG2_DISCOVAR250bp.l100c500.vcf.gz.

De novo assemblies used as inputs to SVRefine were:
PacBio-only:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/UMD_PacBio_Assembly_CA8.3_08252015/

ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/MtSinai_PacBio_Assembly_falcon_03282016/

Illumina-only:

ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/TAMU_NIST_Illumina_2x250bps_DISCOVAR_Assemblies_09162016/

Dovetail-scaffolded assemblies from PacBio and Illumina:

ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/Dovetail_HiRiseScaffolding_10142016/

10x Genomics:

ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/10XGenomics_ChromiumGenome_LongRanger2.2_Supernova2.0.1_04122018/assemblies/

HG004 Illumina 2x250 paired end and 6kb mate-pair sequencing, plus 10x Genomics Chromium linked reads and Bionano optical mapping for scaffolding, using ABySS 1.9, ABySS 2.0, BCALM2, DISCOVARdenovo, Megahit, Minia, SGA, and SOAPdenovo (https://genome.cshlp.org/content/early/2017/02/23/gr.214346.116):

ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/BCGSC_HG004_ABySS2.0_assemblies_12082016/

Vcf files for each genome from each assembly are under:

ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/NHGRI_SVrefine_12062017/

Assemblytics

These callsets, generated independently for each individual in the trio, used assemblytics[63] (version 1.0, code at https://github.com/MariaNattestad/Assemblytics/releases/tag/v1.0) with default parameters and from two haploid de novo assemblies from PacBio. For each genome assembly, the assembly fasta file was aligned to the reference using MUMmer (v3.23) nucmer method with the following parameters: -maxmatch -l 100 -c 500. Assemblytics was run with default parameters (10,000 bp unique sequence anchor length) on the delta file output from nucmer. Results were transformed into VCF format using SURVIVOR[40] and a custom script, and filtered to variants ≥ 20 bp long.

Haploid de novo assemblies used as inputs to assemblytics were fromPacBio-only:

ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/UMD_PacBio_Assembly_CA8.3_08252015/

ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/MtSinai_PacBio_Assembly_falcon_03282016/

MsPAC

These callsets, generated independently for each individual in the trio, used MsPAC v.e30c77e ((https://github.com/oscarlr/MsPAC) with default parameters.[64] PacBio reads aligned to GRCh37 were downloaded from ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG002_NA24385_son/PacBio_Mt

Sinai_NIST/MtSinai_blasr_bam_GRCh37 and phased SNPs were downloaded from ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/10XGenomics_ChromiumGenome_LongRanger2.0_06202016/HG002_NA24385_son/NA24385_GRCh37.vcf.gz. Using the PacBio aligned bam files and 10X phased SNVs as input, MsPAC generated diploid assemblies and phased SVs calls. Assembly fasta/fastq files as wells as VCF files for called SVs can found here: ftp://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/AshkenazimTrio/analysis/MSSM_MsPAC_SVs_assemblies_06042019/

Phased-SV

Haplotype-specific assemblies and associated callsets for HG002 were generated using Phased-SV (github.com/mchaisso/phasedsv) with parameters {"recall_bin": 100, "cov_cutoff": 3, "tr_cluster_size": 6, "depth" : 50}. Reads were aligned to GRCh38 using blasr (github.com/mchaisso/blasr) retaining quality value information. SNP phasing from ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/10XGenomics_ChromiumGenome_LongRanger2.0_06202016/HG002_NA24385_son/NA24385_GRCh37.vcf.gz was used to partition reads by haplotype, and local assemblies were performed using canu.[61] Insertion and deletion SV calls are available at ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/Chaisson_PacBio_smrt-sv.dip_Jun2016/

De novo assemblies

PacBio Canu (haploid)

A non-diploid assembly was generated using the CA 8.3 assembly method[65] from PacBio Continuous Long Read data from each member of the Ashkenazi Trio. The assemblies used MHAP "sensitive" parameters and PBDAGCON for consensus. All assemblies have been polished using Quiver. The assemblies are available at:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/UMD_PacBio_Assembly_CA8.3_08252015/
PacBio data used for each member of the trio is in NCBI SRA SRX1033793–SRX1033798

PacBio Falcon (haploid)

A non-diploid assembly was generated using the Falcon assembly method[18] from PacBio Continuous Long Read data from each member of the Ashkenazi Trio. The assemblies are available at:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/MtSinai_PacBio_Assembly_falcon_03282016/
PacBio data used for each member of the trio is in NCBI SRA SRX1033793–SRX1033798

Illumina DISCOVAR (haploid)

A non-diploid assembly was generated using the DISCOVAR De Novo tool[66] (https://software.broadinstitute.org/software/discovar/blog/) from 2x250bp Illumina sequencing data from each member of the Ashkenazi Trio. The assemblies are available at:

ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/TAMU_NIST_Illumina_2x250bps_DISCOVAR_Assemblies_09162016/
Illumina 2x250bp overlapping read data used for DISCOVAR is in NCBI SRA:
SRX1726837-SRX1726840, SRX1726853-SRX1726856, SRX1726860, SRX1726868, and SRX1726870 for HG002
SRX1726871-SRX1726875 and SRX1726881-SRX1726893 for HG003
SRX1726894-SRX1726928 for HG004

Dovetail Chicago Scaffolding of PacBio and Illumina Assemblies

Dovetail Genomics generated Chicago libraries on HG002, HG003, and HG004 and used HiRise to scaffold 3 existing assemblies for each genome:
1. PacBio Falcon: ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/MtSinai_PacBio_Assembly_falcon_03282016/
2. PacBio PBcR/MHAP: ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/UMD_PacBio_Assembly_CA8.3_08252015/
3. Illumina 2x250bp DISCOVAR: ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/TAMU_NIST_Illumina_2x250bps_DISCOVAR_Assemblies_09162016/
Raw reads are under ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG.../Dovetail_ChicagoLibraties/ for each genome.
Scaffolded assembly results are under each genome ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/Dovetail_HiRiseScaffolding_10142016, with a description of the files in manifest.txt under the hu... directory under each starting assembly. bam files with reads mapped to the assembly are under the bams directory for each genome.

Optical and Electronic Mapping

Bionano

These callsets, generated independently for each individual in the trio, used Bionano Solve v3.1 (bnxinstall.com/solve/Solve3.1_08232017) with default parameters and from Bionano data generated from two enzymes BspQI and BssSI. SV calls and maps are available under:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/BioNano_haplotype_SV_BspQI_BssSI_overlapv0.3.0b_10312017/

Nabsys

This callset, generated for HG002, used Nabsys HD-Mapping with NPS Analysis v1.2.1922 and SV-Verify 12.0.[20] Single molecule reads, ≥50kb were mapped to both GRCh37 and constructs representing putative deletions. Mapping results were evaluated by a set of support vector machines that had been trained on similar sized deletions in NA12878. SV-Verify tests the hypothesis that the specified number of bases, as defined by a putative deletion, are deleted in the region between the lower and upper bound probe locations, specified in the .bed file. Additional SVs (deletions, insertions) occurring within the same region will invalidate the hypothesis. SV-Verify results are under:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/Nabsys_v0.3.0bCorroboration_Sept2017/

Other SV Callsets

We generated additional SV callsets, which were not used in forming or evaluating the v0.6 SV benchmark set, but some were used in previous benchmark versions or are a resource for community evaluations.

Illumina mapping-based TARDIS

These call sets are generated jointly for the trio used TARDIS[67] (version 1.0.4, code at https://github.com/BilkentCompGen/tardis) with default settings from Illumina HiSeq 300x and 100x 2x150bp data for the AJ and Chinese trios. TARDIS SV calls were made from the BAM files and all SVs with read pair support < 50 were filtered out. For the genomes with 100x depth of coverage, the minimum read pair support was 18. TARDIS call sets include deletions, inversions, tandem and interspersed duplications, mobile element insertions, nuclear mitochondrial DNA insertions, and small novel insertions. Only those SVs that are supported by multiple soft-clipped reads are sequence-resolved. The output VCF and BED files are under:

ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/BilkentUni_IlluminaHiSeq_TARDIS_mrCaNaVar_05212019/

ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/BilkentUni_mrCaNaVaR_GRCh38_07242019/

ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/ChineseTrio/analysis/BilkentUni_IlluminaHiSeq_TARDIS_mrCaNaVar_05212019/

Illumina mapping-based MrCaNaVaR

We used mrCaNaVaR tool[68] with default parameters to characterize large (>10 Kb) segmental duplications and deletions, and calculate genic copy numbers. The mrCaNaVaR tool is a reimplementation of an earlier read depth based algorithm designed to detect segmental duplications. Briefly, we remapped the Illumina reads to the repeat-masked reference genome assembly, and identified regions of read depth higher than the genome average (specifically 3 standard deviations above average) after GC% error correction. The output VCF and BED files are under:

ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/BilkentUni_IlluminaHiSeq_TARDIS_mrCaNaVar_05212019/

ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/ChineseTrio/analysis/BilkentUni_IlluminaHiSeq_TARDIS_mrCaNaVar_05212019/

PacBio mapping-based PALMER

We used PALMER (https://github.com/mills-lab/PALMER) to identify non-reference human-specific Long Interspersed Element-1 (L1Hs) insertions and characterize significant hallmarks of these retrotransposon insertions.[69] PALMER firstly pre-masks aligned long-read sequences containing endogenous reference L1Hs and then searches against L1.3 (GenBank Accession: L19088) sequences to detect non-reference L1Hs insertions within the remaining unmasked sequences. After we obtained the preliminary non-reference L1Hs insertions from PALMER, error-correction and local-alignment processes were manipulated by using CANU[65] (https://github.com/marbl/canu) and blasr (github.com/mchaisso/blasr). We run PALMER onto these error-corrected reads and obtained a high

confident set of each sample. The output VCF files, including L1Hs insertion sequences and 5’ and 3’ target site duplicate sequences from PacBio data, are under:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/PacBio_PALMER_11242017/

Detailed Spiral BioGraph Refinement Process

Individual stages can be executed one at a time or the entire process can be run with ‘python Workflow.py all’

## Step 1 - makeDB - Create Database

This removes any existing database and loads the schema in DBSchema.sql into an sqlite3 database named AJTrio.db

Tables:

- GIABVariant
  - Holds a single GIAB Variant as loaded from the union of all variant calls >19bp generated prior to November 11, 2016
- Locus
  - Represents the coordinates of a set of merged GIABVariants
- LocusCalls
  - Holds which GIABVariants are within each Locus
- SpiralVariant
  - Spiral Variant that was created through force calling
- ForceMatch
  - Holds the relation between how each GIABVariant within a Locus was force called with a SpiralVariant
- Trio
  - Matches SpiralVariants across individuals that are identical - representing inheritance.
- Collection
  - Represents the relationship between the elements above. Per locus, we want to show how close any particular GIABVariant is to any Trio Call.

## Step 2 - loadGIAB - Load GIAB Variants

This parses the input GIABVariants and loads them into our GIAB Table.
In order to homogeneously represent all the variants in the database. Some massaging of the data had to be performed.

1. Insertions with identical starts and ends had +1 added to the end to prevent bedtools having problems sorting entries.
2. Chromosome MTT was changed to MT
3. The variant's name entry was split by '_' with properties individual, platform, program, other. However, name property was still preserved
4. Qualities of '.' were translated to -1
5. The end was set with the following logic:
   1. If END was in the info, set to that

    2. else if svtype is INS - set end to start
    3. else if svtype is DEL - set end to start + svlength
    4. if the svtype is one of "BND", "INV_INDEL", we do not set the end coordinate
6. If end was before the start, switch the two values
7. The svtype was set with the following logic:
    1. If SVTYPE, TYPE or SV_TYPE is in info, use that value as the svtype.
    2. Else if the length of the refSeq is greater than the length of the altSeq, it's a deletion
    3. Else if the length of the altSeq is less than the length of the refSeq, it's an insertion
    4. If the svtype is complex - use logic of #2 and #3 above
8. The svlen was set with the following logic:
    1. If SVLEN in info, use that value
    2. else infer the size using altSeq/refSeq or the distance between start and end

## Step 3 - filtGIAB - Filter GIAB Variants

set ignore flags in the GIABVariants table's removed column.

1. if variant's start < 0 : set bad_coord
2. if svtype != ins or del : set bad_type
3. if end-start > MAXSPAN (20k) : set large_span
4. if svlen > MAXLEN (20k) : set large_size
5. If variant intersects with a reference gap : set gap type

## Step 4 - makeLoci - Merge GIAB Variants and Create Loci

Combine the GIABVariants to create loci where we'll be force calling.
Calls were separated by svtype (only DEL and INS were considered)
Used bedtools merge with book-ends distance of MERGEDIST (100)

## Step 5 - cluster - Create Spiral force calls

For every locus, we took a ±300bp buffer from the start and end
If this was ≥ 10kb - we ignored the call.
Resulting force calls were filtered with the following conditions:

1. The variant's annotation is not MNP, SNP, or REF
2. At least 5 reads supported the alternate
3. The variant's length was at least 20bp
4. The variant's annotation starts with the locus' svtype
    1. Note: Many spiral assemblies' remapping on a reference will be a 'net-gain' insertion of 'net-loss' deletion. For example, if 500bp is removed from the sample relative to the reference and replaced with 400bp, this is considered a 100bp DELrp - Deletion with replacement. This example variant can only match with loci with svtype DEL

Each of these force calls are recorded in the SpiralVariant table. The ForceMatch holds what SpiralVariants match to which GIABVariants with metrics.
The logic for matching a the resulting force calls from above is as follows:

1. If force calling failed (couldn’t assemble anything in the region), We add an entry for the locus that is annotated as `ERROR`
2. If there is only a single reference force call, We add that entry into SpiralVariant and annotate the ForceMatch as “REF”
    1. These places may generally be considered “Likely False Positives”

3. If there are no force calls passing the above filtering and more than a single reference force call, No SpiralVariant is recorded and the ForceMatch is annotated as “Unknown”
    1. This could mean a wide variety of things, but the one thing we know for sure is that none of the calls helped us get to a structural variant
4. If there is at least one force call passing the above filtering, for every GIABVariant inside the locus, we choose the one force call with the closest euclidean distance to compare to the variant. We then check if either of the following two conditions are true.
    1. There’s less than 100 euclidean distance from the force call’s start/end and the variant’s start/end
    2. There’s at least 80% reciprocal overlap between the force call and variant positions.
5. If 4.1 or 4.2 are true, we annotate the ForceCall as “Match”, if not, we annotate as “Near”.

Note that in loci with multiple force calls, we only report SpiralVariants in the ForceCall table if they are the nearest to at least one variant.

Supplementary Note 3: Progression of GIAB SV Benchmark Versions

Several draft SV benchmark sets were developed and evaluated by the GIAB community, and feedback from end users and new technologies and SV callers were used to improve each subsequent version. A description of each version is below:

1. v0.2.0 included only deletions ≥20 bp, clustering was performed based only on overlap and size, and the final benchmark was a bed file with deleted regions supported by more than one technology.
2. Users of v0.2.0 requested sequence-resolved calls including insertions, so v0.3.0b included sequence-resolved deletions and insertions ≥20 bp, sequence-based clustering was performed by SVanalyzer (and experimentally by breakpoint using SURVIVOR in v0.3.0a), and the final v0.3.0b benchmark was a vcf file with sequence-resolved insertions and deletions supported by more than one technology. For v0.3.0b, only callsets with sequence-resolved calls were used. This version is under ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/NIST_UnionSVs_05092017/Preliminary_Integrations_v0.3.0/
3. Users of v0.3.0b found that few large insertions were included due to the requirement of support from multiple technologies, and only long reads discovered large insertions. In addition, some errors in v0.3.0b resulted from short and linked read mis-assemblies. Therefore, v0.4.0 used the same input callsets and clustering methods as v0.3.0b, and used svviz to evaluate the support for each variant by short, linked, and long reads, as well as the genotype of the SV in HG002. Calls were included in v0.4.0 if they had consistent genotypes even if they were discovered by only one technology, and calls supported by Bionano or Nabsys were also included. This version is under ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/NIST_UnionSVs_05092017/Preliminary_Integrations_v0.4.0/
4. Users of v0.4.0 reported that very large insertions, particularly LINEs, were still missing from the benchmark. Therefore, several improvements were made in v0.5.0:
    a. For the input callsets, especially to improve calling of large insertions, we have added:
        i. Updated sequence-resolved svanalyzer calls from assemblies, using a new version includes many more large insertions, run in discovery mode instead of targeted mode, merged calls from unscaffolded and scaffolded assemblies into single callset, and added diploid PacBio assembly from Mt Sinai.
        ii. New sequence-resolved BreakScan calls from NCBI
    b. Sites were genotyped in each member of the trio.
    c. To increase sensitivity, we excluded fewer single-tech candidate calls, requiring discovery by 4 callsets across the trio (v0.4.0 required 5 callsets in an individual)
    d. To output a heterozygous or homozygous variant GT for an individual, we required that the call was discovered in that individual (BioNano included). This eliminated some cases where the variant predicted for an individual was significantly different from the true variant in that individual.
    e. We included svviz 2.0 (https://github.com/nspies/svviz2) analysis with PacBio bam separated by haplotype using whatshap with 10X variants, and we output local phasing from PacBio and 10X when available.
    f. This version is under ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/NIST_UnionSVs_12122017/

5. Users of v0.5.0 reported that it was useful for assessment of FNs, but not FPs, and that about a third of the v0.5.0 SVs were potentially complex or inaccurate, since they were within 1000 bp of another v0.5.0 SV. Therefore, in v0.6 we separated the calls into 2 tiers: (1) isolated, sequence-resolved SVs and (2) regions with at least one likely SV but it is complex or we were unable to determine a consensus sequence change. We also created the first SV benchmark bed file, defining regions in which the Tier 1 callset should be comprehensive, so that any extra variants detected by a method should be false positives. We also focused this benchmark to include only calls ≥50 bp and only calls in HG002, though genotypes for the parents are provided as annotations. The methods used to create v0.6 are described in this manuscript. This version is under ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/NIST_SVs_Integration_v0.6/

Supplementary Note 4: Evaluation of the Benchmark

Truvari benchmarking tool used in evaluation

For the evaluations, we used the tool truvari (https://github.com/spiralgenetics/truvari). This tool takes the benchmark vcf, benchmark bed file, and query vcf file as inputs. For each call in the benchmark vcf within the benchmark bed, it compares all calls in the query vcf within a user-specified distance (2000 bp for our comparisons). It picks only the best match, unless --multimatch is specified. To be counted as matching, the ratio of benchmark and query size estimates must be within 30 %, since we specified pctsize=0.7. For some of our benchmarking evaluations (e.g., for the SVRefine calls), we further specified that the Levenstein distance between the sequences in the benchmark and query vcfs must be less than 30 % of the size of the SV (pctsim=0.7). Outputs from each of the benchmarking evaluations and log files with parameters used are under:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/NIST_SVs_Integration_v0.6/GIAB_Evaluations/

Callsets benchmarked against v0.6 Tier 1 benchmark set

Illumina mapping-based Delly and Manta

Structural variants were called from Illumina HiSeq 300x 2x150bp data (previously aligned to hs37d5) using Manta (version 1.2.2, code at https://github.com/Illumina/manta) with default options and Delly (version 0.7.8, code at https://github.com/dellytools/delly) with minimum mapping quality set to 20. For Manta, all calls from diploidSV.vcf output file were filtered for the "PASS" filter field. For Delly, SVs were discovered, merged with Delly's default values (breakpoint offset: 1000 & reciprocal overlap: 0.8) and genotyped. Output calls were filtered according to the "PASS" filter and a minimal count of alternate-supporting reads of 5. SVs in centromeres and telomeres were excluded, with the list provided with Delly developers. Calls were compared to the goldset using truvari, and manually verified in IGV.

Illumina mapping-based MetaSV

The same MetaSV callset described above was benchmarked against v0.6.

Illumina assembly-based SpiralBGA

The same Spiral BioGraph methods described above was benchmarked against v0.6

PacBio mapping-based pbsv

PacBio 10 kb CCS reads (ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG002_NA24385_son/PacBio_CCS_10kb/) were aligned to hs37d5 with minimap2 version 2.11-r797 (https://github.com/lh3/minimap2).[70] Structural variants were called with pbsv version 2.0.0 (https://github.com/PacificBiosciences/pbsv) with default parameters. Variants were evaluated against the GIAB v0.6 benchmark set using Truvari commit bb51e7575 with "--passonly --pctsim 0 -r 2000 --giabreport". Ten randomly selected variants were evaluated in IGV for each combination of variant type (insertion, deletion); Truvari error type (false positive, false negative); and overlap with tandem repeats (yes, no). The pbsv VCF files are at:

ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/PacBio_pbsv_05212019/

PacBio assembly-based SVRefine from MsPAC Diploid Assembly

The following vcfs were combined with svanalyzer, merging calls within 20% edit distance, and compared to the v0.6 Tier 1 benchmark with truvari:

ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/NHGRI_SVrefine_05252018/hs37d5/HG2_ORod_raw.01_0518.l100c500.no_ns.vcf.gz

ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/NHGRI_SVrefine_05252018/hs37d5/HG2_ORod_raw.02_0518.l100c500.no_ns.vcf.gz

These vcfs were generated from each assembled haplotype of MsPAC[60] in the folder below using SVRefine, as described above:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/MSSM_MsPAC_SVs_assemblies_06042019/

10x Genomics mapping-based LongRanger

This callset used LongRanger (version 2.2, code at https://github.com/10XGenomics/longranger) with default parameters and 10x Genomics data for HG002 (86x). Indels >19bp from *_phased_variants.vcf.gz and large deletions from *_deletions.vcf.gz were converted into sequence-resolved vcf format. Vcf and bam files for each genome are under:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/10XGenomics_ChromiumGenome_LongRanger2.2_Supernova2.0.1_04122018/

Bionano Genomics

GM24385 data generated using the DLS chemistry and Bionano Saphyr system are assembled and have SVs called against hg19 using Bionano Solve v3.2.2 (bnxinstall.com/solve/Solve3.2.2_08222018)) with default parameters. The data and SV calls are under:
ftp://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/analysis/BioNano_haplotype_SV_06072019

Bionano indels (>1 kbp) are overlapped with the 1557 v0.6 benchmark calls (>1 kbp) that showed “PASS” in the FILTER column; Bionano calls with 80% size concordance and 50% reciprocal position overlap are clustered to avoid duplicated counts of homozygous calls. Between the Bionano calls and the v0.6 benchmark calls, a size concordance of 50% and breakpoint precision of 5kbp are required for them to be overlapped. About 90% of the v0.6 benchmark calls overlapped with Bionano, and an additional 1 % match when summing of neighboring indels within the same Bionano markers. Only a few Bionano unique calls fall within the Tier1 region, but over a thousand Bionano unique calls fall outside of the Tier1 regions, where Bionano may be able to detect SVs in repetitive regions spanned by its ultra-long (323 kbp N50) molecules. Future work will be needed to develop robust benchmarks for complex events and very large, repetitive regions.

Supplementary References

50. Šošić, M. & Šikić, M. Edlib: a C/C++ library for fast, exact sequence alignment using edit distance. *Bioinformatics* **33**, 1394–1395 (2017).
51. Martin, M. *et al.* WhatsHap: fast and accurate read-based phasing. *bioRxiv* 085050 (2016) doi:10.1101/085050.
52. Garg, S., Martin, M. & Marschall, T. Read-based phasing of related individuals. *Bioinformatics* **32**, i234–i242 (2016).
53. Iqbal, Z., Caccamo, M., Turner, I., Flicek, P. & McVean, G. De novo assembly and genotyping of variants using colored de Bruijn graphs. *Nat. Genet.* **44**, 226–232 (2012).
54. Chen, X. *et al.* Manta: rapid detection of structural variants and indels for germline and cancer sequencing applications. *Bioinformatics* **32**, 1220–1222 (2016).
55. McKenna, A. *et al.* The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res.* **20**, 1297–1303 (2010).
56. Garrison, E. & Marth, G. Haplotype-based variant detection from short-read sequencing. (2012).
57. Li, H. FermiKit: assembly-based variant calling for Illumina resequencing data. *Bioinformatics* **31**, 3694–3696 (2015).
58. Mohiyuddin, M. *et al.* MetaSV: an accurate and integrative structural-variant caller for next generation sequencing. *Bioinformatics* **31**, 2741–2744 (2015).
59. Freed, D., Pan, R. & Aldana, R. TNscope: Accurate Detection of Somatic Mutations with Haplotype-based Variant Candidate Detection and Machine Learning Filtering. *bioRxiv* (2018).
60. Narzisi, G. *et al.* Accurate de novo and transmitted indel detection in exome-capture data using microassembly. *Nat. Methods* **11**, 1033–1036 (2014).
61. Rakocevic, G. *et al.* Fast and accurate genomic analyses using genome graphs. *Nat. Genet.* **51**, 354–362 (2019).
62. Fan, X., Chaisson, M., Nakhleh, L. & Chen, K. HySA: a Hybrid Structural variant Assembly approach using next-generation and single-molecule sequencing technologies. *Genome Res.* **27**, 793–800 (2017).
63. Nattestad, M. & Schatz, M. C. Assemblytics: a web analytics tool for the detection of variants from an assembly. *Bioinformatics* **32**, 3021–3023 (2016).
64. Rodriguez, O. L., Ritz, A., Sharp, A. J. & Bashir, A. MsPAC: A tool for haplotype-phased structural variant detection. *Bioinformatics* (2019) doi:10.1093/bioinformatics/btz618.
65. Koren, S. *et al.* Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation. *Genome Res.* **27**, 722–736 (2017).
66. Weisenfeld, N. I. *et al.* Comprehensive variation discovery in single human genomes. *Nat. Genet.* **46**, 1350–1355 (2014).
67. Soylev, A., Le, T., Amini, H., Alkan, C. & Hormozdiari, F. Discovery of tandem and interspersed segmental duplications using high throughput sequencing. *Bioinformatics* (2019) doi:10.1093/bioinformatics/btz237.
68. Alkan, C. *et al.* Personalized copy number and segmental duplication maps using next-generation sequencing. *Nat. Genet.* **41**, 1061–1067 (2009).
69. Zhou, W. *et al.* Identification and characterization of occult human-specific LINE-1 insertions using long-read sequencing technology. *Nucleic Acids Research* **48**, 1146–1163 (2020) doi:10.1093/nar/gkz1173
70. Li, H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* **34**, 3094–3100 (2018).