Application Notes

SVA: Software for Annotating and Visualizing Sequenced Human Genomes

Supplementary Information

Materials and Methods

Resources used by SVA

SVA integrates and compiles data from the following resources for performing the analyses. These resources are compiled into one package and are released with SVA. We will periodically release new data resources and notify SVA users through the SVA version control system.

1. NCBI RefSeq: The Reference Sequence (RefSeq) collection aims to provide a comprehensive, integrated, non-redundant, well-annotated set of sequences, including genomic DNA, transcripts, and proteins. RefSeq provides a stable reference for genome annotation, gene identification and characterization, mutation and polymorphism analysis, expression studies, and comparative analyses(Pruitt, et al., 2007).
2. Ensembl databases: Ensembl (Flicek, et al., 2010) is a joint project between EMBL - EBI and the Wellcome Trust Sanger Institute. It provides a system that produces and maintains automatic annotation on selected eukaryotic genomes.
3. NCBI dbSNP: The dbSNP database is a publicly available database of genetic variation and associated information. It is integrated with other sources of information such as GenBank, PubMed, LocusLink and the Human Genome Project data (Sherry, et al., 2001).
4. HapMap: The International HapMap Project (Frazer, et al., 2007) is a multi-country effort to identify and catalog genetic similarities and differences in human beings.

5. Illumina Infinium HD Human1M BeadChip: a commercial GWAS (genome-wide association studies) genotyping chip manufactured by Illumina Inc (http://www.illumina.com/products/human1m_duo_dna_analysis_beadchip_kits.ilmn).
6. DGV: A curated catalogue of structural variation in the human genome(Iafrate, et al., 2004).
7. HuRef: The diploid genome sequence of J. Craig Venter(Levy, et al., 2007).
8. The 1000 Genomes Project: The 1000 Genomes Project is an international research effort to establish a detailed catalogue of human genetic variation(Durbin, et al., 2010).
9. KEGG pathway: KEGG pathway is a collection of manually drawn pathway maps representing knowledge on the molecular interaction networks for certain biological processes(Kanehisa, et al., 2010).
10. Gene ontology: The Gene Ontology project aims to standardize the representation of gene and gene product attributes across species and databases (Ashburner, et al., 2000).
11. OMIM: Online Mendelian Inheritance in Man. OMIM is a database of information on known Mendelian disorders and disease related genes. OMIM is a compilation of known relationships between phenotype and genotype(http://www.ncbi.nlm.nih.gov/omim/, 2009).
12. RepeatMasker: RepeatMasker is a program that screens DNA sequences for interspersed repeats and low complexity DNA sequences(Smit, et al., 2010).
13. PROBCONS: Probabilistic Consistency-based Multiple Alignment of Amino Acid Sequences(Do, et al., 2005).
14. SEMPHY: Structural EM Phylogenetic Reconstruction(Friedman, et al., 2002).
15. MAPP: Multivariate Analysis of Protein Polymorphism(Stone and Sidow, 2005).

Results

Overview of SVA

We have developed a computational environment to analyze genetic variants identified in next-generation sequencing studies. This environment utilizes a knowledgebase of 8.9GB, which is compiled and compressed into one DVD ROM or can be downloaded from the SVA website, to systematically combine the information from a great number of biological databases. Human association genetics using whole-genome sequence data as opposed to GWAS data has a key fundamental difference. Whereas it was appropriate to treat all variants as equally likely to show a real association in GWAS (since the tested variants were primarily markers for a large number of unknown variants), sequencing identifies specific variants that often have known biological functions. For example, an identified variant that results in protein truncation, or in an amino acid substitution, would appropriately be assigned a somewhat high *a priori* probability of influencing the studied phenotype than a variant that is far from any annotated feature of the human genome. In order to help researchers capitalize on this information, SVA attempts to organize all identified variants into specific groups based on whether they fall in genomic regions that are annotated as clearly functional. This annotation of variants provides the means to interpret and prioritize the significance of variation in the interrogated human genome, and provides a smooth interface between association evidence and the bioinformatic annotation of identified variants.

We now describe in detail the main modules in SVA.

The Annotation Module

Using a highly cross-referenced knowledgebase, the SVA annotation module performs three main functions (Figure S1): (1) classifies each variant into relevant functional categories; (2) checks for each variant in existing databases; and (3) enables the filtering of variants. The annotation results can then be

displayed and navigated in a built in genome browser (Figure S5). In addition, quality scores and coverage data can be summarized, displayed, and filtered .

Different researchers utilize different next-generation sequencing strategies and use any number of alignment and variant calling software programs to generate their NGS dataset. With minimal data conversions, SVA can use these identified variants as its input and thus SVA can be added to the end of existing sequencing pipelines. Therefore, SVA's flexibility enables researchers to determine the importance of these variants *regardless* of the research strategy implemented upstream of this variant annotation.

(1) Functional Categories of Variants

The SVA annotation module firstly uses the variant coordinate and allele information to match the variants identified by sequencing with the RefSeq (Pruitt, et al., 2007) and Ensembl (Flicek, et al., 2010) databases. This provides a genomic context for each of the identified variants. Additionally, the RepeatMasker program (Smit, et al., 2010) is used to screen for known repetitive regions. Based on this genomic context, each identified variant is assigned to one or more *functional categories* (Table S1, e.g., nonsynonymous coding or 3'UTR). It should be noted that a variant might be assigned to the same functional category multiple times, or to multiple distinct functional categories because it resides in several transcripts belonging to the same gene or several different overlapping genes.

(2) Identifying Novel Variants: Searching Existing Databases

In order to determine the "novelty" of an identified genetic variant (SNV or indel), each is compared to the refSNP (dbSNP) database. If a dbSNP entry and an rs number exist for an identified variant, the variant is labeled with this rs number. Similarly, if structural variants have been called, these are compared to the exact allelic coordinates listed in the database of genomic variants (DGV) (Iafrate, et al., 2004). In addition, the variants are compared to HapMap (Frazer, et al., 2007), the 1000 Genomes Project

(Durbin, et al., 2010), the HuRef sequence (Levy, et al., 2007) and are checked for their presence on Illumina's 1M genome-wide genotyping BeadChip.

(3) Variant or Gene Filtering

The third main function of the annotation module enables the user to filter genetic variants. SVA has two types of filters: those that act on a genomic region, gene, or gene-set (e.g., gene ontology terms, OMIM disorders, or KEGG pathways) and those that are specific to the other properties of a variant (e.g., quality, number of reads supporting a variant, or genotype).

Annotation Results and Visualization Module (SVA Genome Browser)

The annotation module produces two classes of output files: 1) *the functionally annotated dataset* (*.sva* file, binary) and 2) *the genome browser dataset* (*.gbb* or *.gb2* file, binary). In addition, SVA is able to export the annotated dataset into flat text files, which can be saved and used for external analyses. The user can navigate SVA's genome browser by chromosomal coordinate, gene symbol, refSNP, or by zooming manually. The genome browser displays: repetitive regions, genes, variants identified in each subject genome, variants from public databases, and also information about the quality of the sequence in each genome at a given location (Figure S5). The genome browser is created by overlaying the annotated variant dataset with the *.gbb* or *.gb2* file – which contains information for the genome browser. The genome browser file (*.gbb* or *.gb2*) only needs to be generated once and can be shared among multiple SVA projects as long as the version of the genome build and the number of chromosomes with sequence data are the same. Referencing an existing *.gbb* or *.gb2* file speeds annotation and is accomplished by providing the *.gbb* file location in the project script file (*.gsap*).

Simple listing functions

SVA provides a number of simple bioinformatic listing functions for prioritizing genetic variants. For example, one application would be to list all variants in specific functional categories that are present in one or more cases and entirely absent in controls. Such listing functions can be of particular relevance in studies where sample sizes are very limited, as we illustrate later using the example of metachondromatosis. The listing functions rank all variants satisfying particular criteria on the basis of the degree of the enrichment in the case genomes. If there are multiple variants fulfilling these criteria, additional filtering could be used to narrow this list, such as linkage regions, gene function, or known biological pathways.

When these listing functions are combined with any number of the filters available in SVA, the power of customization is greatly increased. For example, if the user were analyzing epilepsy genomes, it might make sense to make use of the fact that many of the known epilepsy genes are ion channels. This phenotype-specific information could be "input" by a applying a filter for the GO term "ion channel", prior to running any analyses. This would greatly narrow the list of candidate variants in a given epilepsy genome.

Implementation and Performance

A 64-bit JAVA environment is required. To accommodate the graphical user interface of SVA, this workstation also needs to support a Graphical User Interface (GUI) environment, for example an X-windows system. SVA is a JAVA program so it is platform independent. We recommend SVA be run on a 64-bit Linux workstation equipped with 48GB of RAM or more. There are also options to reduce the system requirement. SVA also provides a command line tool that can be implemented into an NGS pipeline. To run this command line tool, no GUI is required. We also provide an evaluation version of SVA which uses an example project (only one chromosome from 10 genomes) that can run on a 32-bit laptop.

We evaluated the computational performance of SVA on a Linux workstation (model: Dell Precision T7500, OS: CentOS, RAM: 96GB, CPU: Intel Xeon CPU 3.33GHz two quad-cores). SVA performance was evaluated using whole-genome and whole-exome sequencing data, generated in our center using Illumina GAIIx sequencing machines, from 10 to 60 unique genomes. Figure S6 summarizes this evaluation of SVA performance. Considering the fact that alignment and variant calling of NGS data from a single genome requires tens to hundreds of CPU hours (depending on data sizes and available computational resources), we consider the computational performance of SVA to be both efficient and relatively easy to implement in separate laboratory settings.

Inputs and Outputs

For each genome, SVA requires the input of genetic variants emerging from next-generation sequencing data in three separate files. These files contain: 1) identified single nucleotide variants (SNVs), 2) identified insertions/deletions (indels), and 3) sequence coverage and, if available, quality control information. The user may choose to include a fourth, optional input file for analyzing structural variations (SVs). Copy number variations (CNVs) can be identified using any available method (Bentley, et al., 2008; Zhu, et al., *Manuscript in progress*) and can then be input into SVA. There are several necessary procedures for processing raw sequence data to create input files that are recognized by SVA.

The input of NGS data in a unified variant format enables the simultaneous analysis of genomes, even if they are sequenced on different sequencing platforms. Additionally, SVA can analyze data from multiple types of NGS projects, including whole-genome, whole-exome, or region captured sequencing. These data sets can also be readily compared with one another, for example by the bioinformatic "masking" of non-coding sequence in whole-genome sequence data to allow only genomically matched comparisons with whole-exome data sets.

Older SVA versions (prior to version 1.1) supported the input of identified variants in the Samtools pileup file format (Li, et al., 2009). The current SVA version 1.1 provides additional supports the VCF (Variant Call Format) data files, defined by the 1000 Genomes Project (Durbin, et al., 2010) and generated from a number of variant calling software tools including Samtools, as the main input. In short, VCF is a text file format. It contains meta-information lines, a header line, and then data lines each containing information about a position in the genome. Users may find the detailed description of this format on the 1000 Genomes Project's website (http://www.1000genomes.org/node/101, 2011). Lastly, we emphasize that the primary features of SVA are robust to whichever sequencing platform or variant calling routines are utilized. As they develop, new sequencing platforms and input files will be incorporated into the SVA workflow.

The primary output of SVA is a set of indexed binary files, storing the analyzed genetic variant data. These outputs can then be conveniently exported into flat text files. A project script file (text format with file name extension *.gsap*) organizes the user inputs, outputs, and the updateable annotation databases released with SVA. Multiple genomes can be analyzed simultaneously within the same SVA project. The SVA project script file is created to define: which genomes to include, basic phenotypic information (case vs. control), the location of processed sequence data files for these genomes, the location and version of annotation databases, and customizable parameters for annotation.

Sequencing Coverage and Quality Scores

Once SVA has compiled all of the alignment and variant data, it is also able to summarize the coverage and variant quality scores across the entire genome. It summarizes the overall coverage and exome coverage as well as the coverage across each chromosome. The annotation output also links various quality score measures, when available, to each corresponding variant identified in the sequencing data. This quality score information is used by SVA to filter for high confidence variants and aims to reduce false positives from the variant dataset.

Annotation: Functional Categories

SVA uses various *functional categories* to help classify the genetic variants. Table S1 lists a complete description of these categories. Each identified variant is assigned to one or more functional categories because of a) inclusion in multiple transcripts or b) an overlap within the functional categories. A variant might be assigned to the same functional category multiple times, or to multiple distinct functional categories because it resides in several transcripts belonging to the same gene or several different genes. Alternately, a variant may be assigned to multiple categories because of the overlap of functional definitions. For instance, the intron-exon boundary often contains variants that will also be classified in a second functional category such as splice site, essential splice site, or nonsynonymous coding. The variant tables in SVA (Figure S4) list all classifications for a given variant but it lists the "most obviously functional" variant first. In addition, the RepeatMasker program (Smit, et al., 2010) is used to screen for known repetitive regions.

Project Script File

To assemble all SVA input files, a simple user-constructed SVA project script file is created to define: which genomes are included, basic phenotypic information, the location of processed sequence data files for these genomes, the location and version of annotation databases, and customizable parameters for annotation. This project script file (text format with file name extension *.gsap*) organizes these user inputs together with the updateable annotation databases released with SVA (Figure S8). Multiple genomes can be analyzed simultaneously within the same SVA "project". Furthermore, to facilitate variant analyses the project script file also supports simple phenotypic information about the included genomes. Currently, each genome can be assigned as either a case or a control (to allow frequency comparisons, for example between individuals with and without a disease). Individual SVA projects are customizable for various parameters, including details that define an annotated functional category (e.g., number of base pairs designated "upstream" of a gene), version control for databases (e.g., NCBI reference genome build used), and restriction of annotation to specific

genomic regions (e.g., annotation of only selected exonic regions for whole-exome sequence data). Editing of the project script file (*.gsap*) accomplishes this customization.

Variant Tables

The SVA variant tables (Figure S4) can be exported to flat text files (.csv). These can be used in combination with any of the possible filter options. For each variant, the number of total subjects carrying a variant is listed and additional subject details, including genotype (homozygous vs. heterozygous) and quality scores, can be obtained by clicking this number. Both cases and controls are listed in the same table and a column entitled "in control" is included for easy determination of control status.

Variant or Gene Filtering

Many useful filters can be applied at the "gene level", including: gene, genomic region, gene biotype, gene ontology term, KEGG pathway (Kanehisa, et al., 2010), and OMIM disorder associations. The *genomic region filter* allows the use of chromosomal coordinates for any regions of interest and obtains only variants falling within the specified coordinates. The genomic region filter is especially tractable; any genomic region of interest can be entered as the input in a tab delimited format indicating the chromosome, start coordinate, and end coordinate. The coupling of this filter with any of the analysis functions provides a way to interrogate genomic regions of interest. This allows the user to focus on linkage regions, regulatory regions, or other known regions of interest. The *gene filter* allows the selection of genes from an interactive list of gene symbols (HUGO gene nomenclature committee). The *gene biotype filter* allows the selection on biological types of genes (e.g., protein coding, pseudogene, miRNA, rRNA) to include. The filters for *gene ontology*, *KEGG pathway*, and *OMIM* select genes based on their known biological functions.

Many useful filters can be applied at the "variant level", including: refSNP, HapMap frequency, intolerable non-synonymous (NS), repeat regions (RMR), functional category, and quality filtering. The *refSNP filter* allows the user to include or exclude variants classified as "novel". The *RMR filter* allows

users to include or exclude variants in known repetitive regions (RepeatMasker program). The *SNP function filters* allow the user to look at variants within a specified functional category (Table S1). Finally, *quality filtering* can be used to filter variants based on their sequencing quality.

Options to reduce memory requirement

In general, for best efficiency we recommend SVA be run on a 64-bit Linux workstation equipped with 48GB of RAM or more. There are several options available to reduce the memory requirement, including: (i) reducing the number of computing threads (simultaneous runs administrated by SVA); and/or (ii) working on chromosome-wise projects, or data splits; and/or (iii) omitting/masking unwanted annotation databases. Users may opt to add a statement of "[MINIMUMMEMORY] = ON" in their .gsap script file to apply the minimum memory usage settings. In addition, users also have the option of utilizing publicly available computational resources (e.g., Amazon Elastic Compute Cloud) at very low cost.

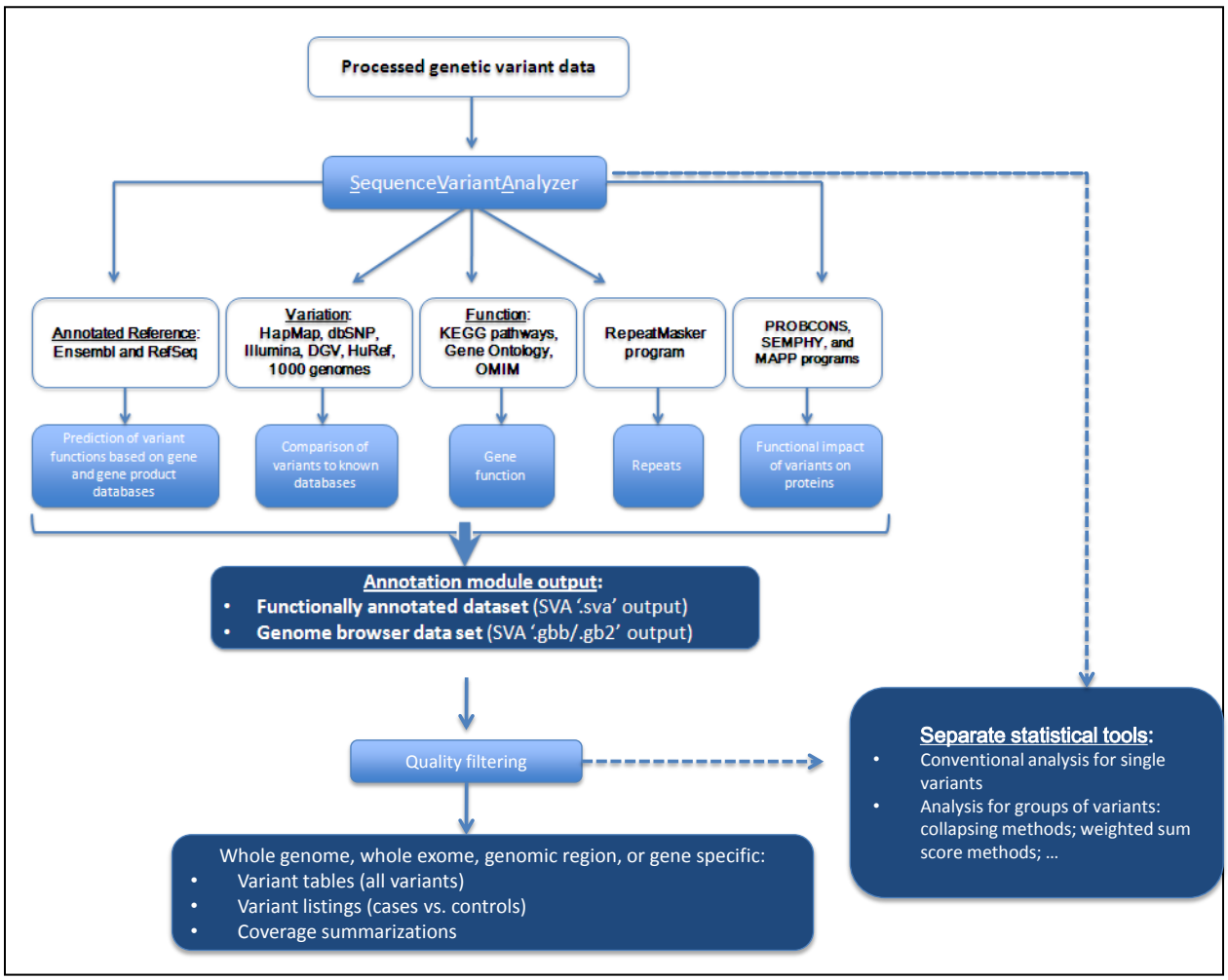


Figure S1. Diagram of the dataflow. Solid lines indicate steps included in the annotation module and dashed lines indicate steps included in the following analysis steps.

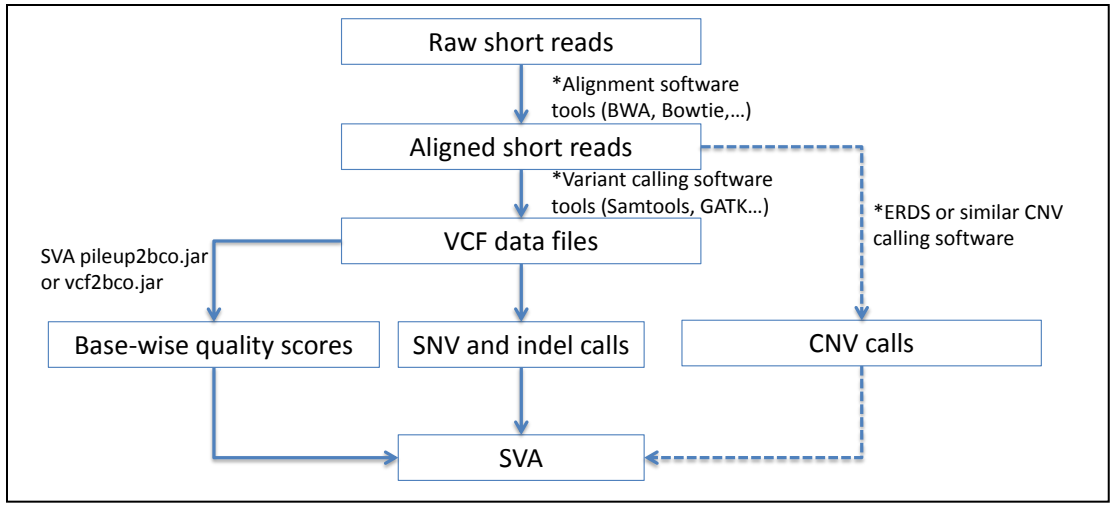


Figure S2. Data input for SVA. Asterisks indicate supplemental programs for SVA input preparation. Several JAVA programs (pileup2bco.jar, or vcf2bco.jar) and PERL scripts accompany SVA and can be downloaded from http://www.svaproject.org/. These variant calls should be in the VCF file format or SAMtools pileup file format (for SVA versions prior to 1.1). The dashed line indicates the optional input of data (tab delimited format indicating the chromosome, start coordinate, and end coordinate) for copy number variations (CNVs), or structural variations (SVs).

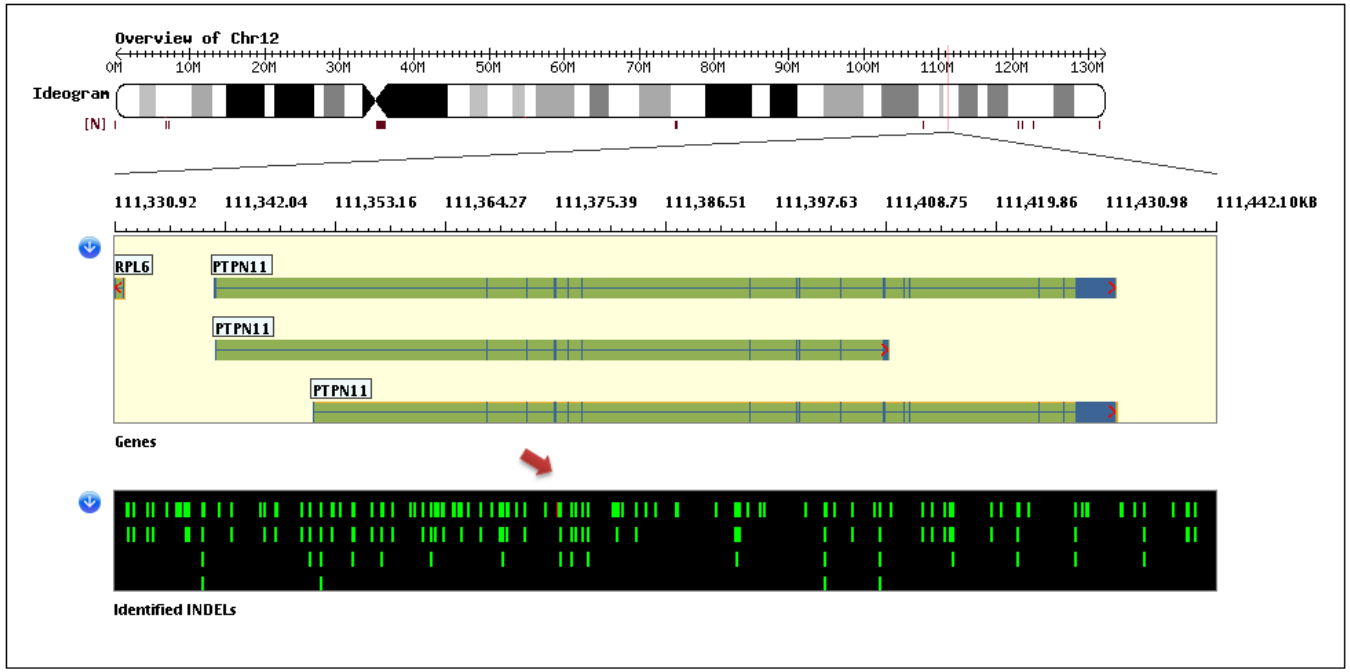
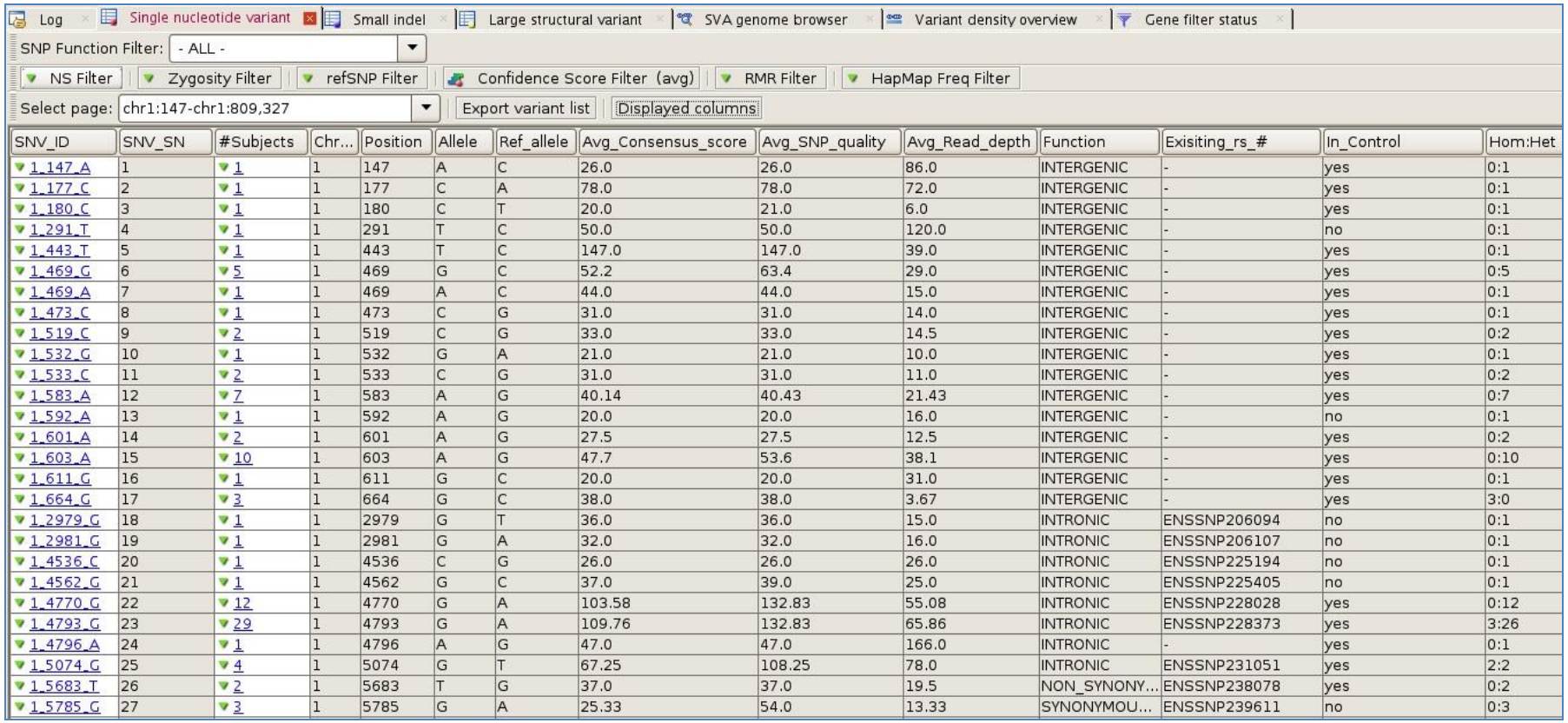


Figure S3. SVA shows that an 11-bp deletion is the causal genetic variant for metachondromatosis (red arrow).



SNV_ID	SNV_SN	#Subjects	Chr...	Position	Allele	Ref_allele	Avg_Consensus_score	Avg_SNP_quality	Avg_Read_depth	Function	Exisiting_rs_#	In_Control	Hom:Het
1_147_A	1	1	1	147	A	C	26.0	26.0	86.0	INTERGENIC	-	yes	0:1
1_177_C	2	1	1	177	C	A	78.0	78.0	72.0	INTERGENIC	-	yes	0:1
1_180_C	3	1	1	180	C	T	20.0	21.0	6.0	INTERGENIC	-	yes	0:1
1_291_T	4	1	1	291	T	C	50.0	50.0	120.0	INTERGENIC	-	no	0:1
1_443_T	5	1	1	443	T	C	147.0	147.0	39.0	INTERGENIC	-	yes	0:1
1_469_G	6	5	1	469	G	C	52.2	63.4	29.0	INTERGENIC	-	yes	0:5
1_469_A	7	1	1	469	A	C	44.0	44.0	15.0	INTERGENIC	-	yes	0:1
1_473_C	8	1	1	473	C	G	31.0	31.0	14.0	INTERGENIC	-	yes	0:1
1_519_C	9	2	1	519	C	G	33.0	33.0	14.5	INTERGENIC	-	yes	0:2
1_532_G	10	1	1	532	G	A	21.0	21.0	10.0	INTERGENIC	-	yes	0:1
1_533_C	11	2	1	533	C	G	31.0	31.0	11.0	INTERGENIC	-	yes	0:2
1_583_A	12	7	1	583	A	G	40.14	40.43	21.43	INTERGENIC	-	yes	0:7
1_592_A	13	1	1	592	A	G	20.0	20.0	16.0	INTERGENIC	-	no	0:1
1_601_A	14	2	1	601	A	G	27.5	27.5	12.5	INTERGENIC	-	yes	0:2
1_603_A	15	10	1	603	A	G	47.7	53.6	38.1	INTERGENIC	-	yes	0:10
1_611_G	16	1	1	611	G	C	20.0	20.0	31.0	INTERGENIC	-	yes	0:1
1_664_G	17	3	1	664	G	C	38.0	38.0	3.67	INTERGENIC	-	yes	3:0
1_2979_G	18	1	1	2979	G	T	36.0	36.0	15.0	INTRONIC	ENSSNP206094	no	0:1
1_2981_G	19	1	1	2981	G	A	32.0	32.0	16.0	INTRONIC	ENSSNP206107	no	0:1
1_4536_C	20	1	1	4536	C	G	26.0	26.0	26.0	INTRONIC	ENSSNP225194	no	0:1
1_4562_G	21	1	1	4562	G	C	37.0	39.0	25.0	INTRONIC	ENSSNP225405	no	0:1
1_4770_G	22	12	1	4770	G	A	103.58	132.83	55.08	INTRONIC	ENSSNP228028	yes	0:12
1_4793_G	23	29	1	4793	G	A	109.76	132.83	65.86	INTRONIC	ENSSNP228373	yes	3:26
1_4796_A	24	1	1	4796	A	G	47.0	47.0	166.0	INTRONIC	-	yes	0:1
1_5074_G	25	4	1	5074	G	T	67.25	108.25	78.0	INTRONIC	ENSSNP231051	yes	2:2
1_5683_T	26	2	1	5683	T	G	37.0	37.0	19.5	NON_SYNONY...	ENSSNP238078	yes	0:2
1_5785_G	27	3	1	5785	G	A	25.33	54.0	13.33	SYNONYMOU...	ENSSNP239611	no	0:3

Figure S4. SVA's Single Nucleotide Variant table. The user can specify which columns are visible and a subset of the columns have been selected here. The tabs seen along the top of this table provide access to similar screens for other types of variants (small indels or large structural variants), as well as other screens for viewing genomic data or project information.

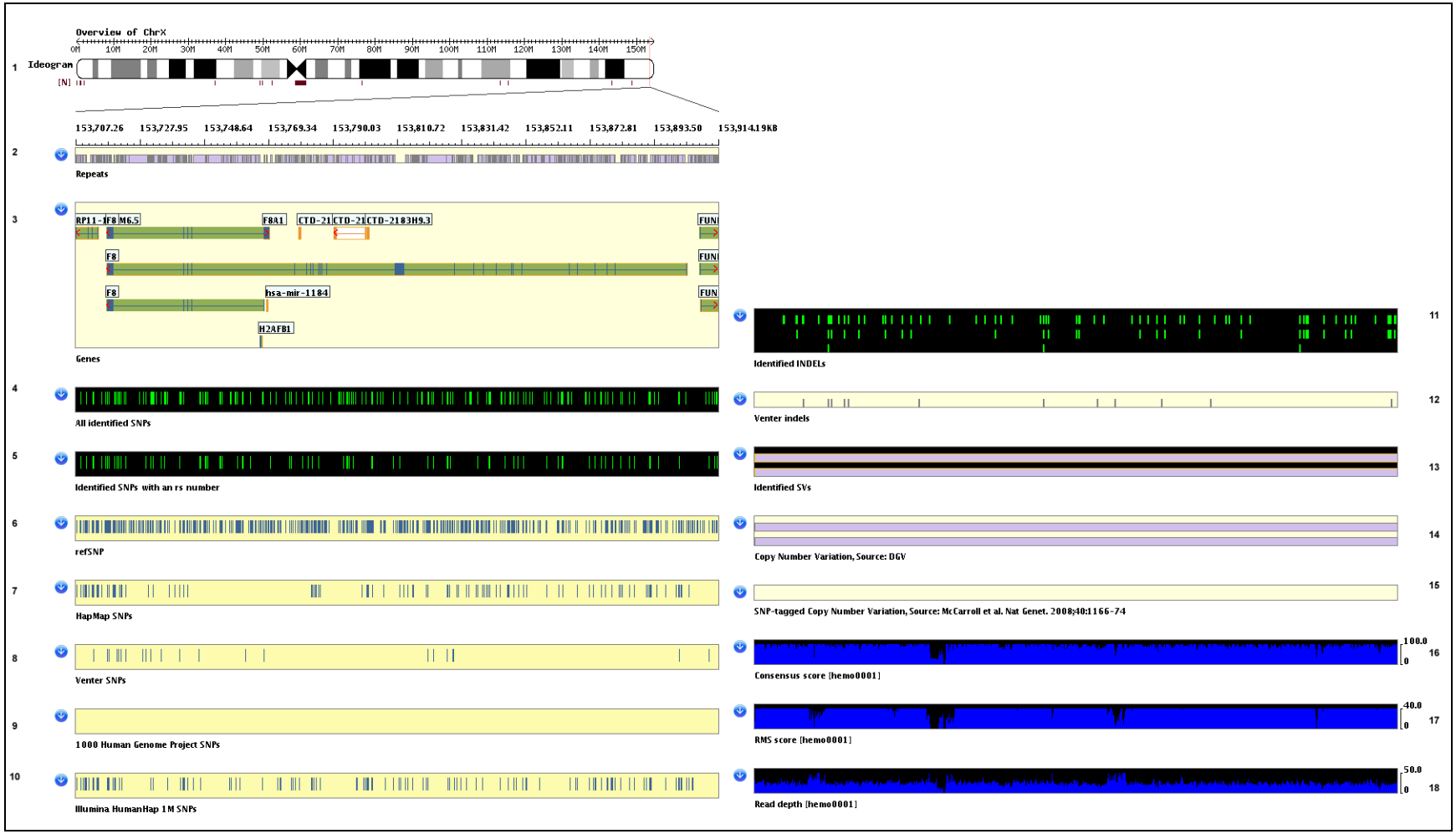


Figure S5. SVA genome browser. The original vertical screenshot is broken down into left and right parts for better visualization.

Displayed are: Left: (1) An ideogram of chromosome X; (2) Genomic coordinates with repeat regions; (3) Genes and their transcripts in this region. Exons are plotted as blue vertical lines/rectangles. Introns are plotted as green rectangles; (4) Identified SNVs (green vertical lines) in this

region; (5) Identified SNVs with an existing rs number; (6) RefSNP entries (blue vertical lines); (7) HapMap SNP entries; (8) SNP entries in Craig Venter's genome; (9) SNP entries in the 1000 human genome project; (10) SNP entries in the Illumina 1M beadchip.

Right: (11) Identified indels (green vertical lines) in this region; (12) Indel entries in Craig Venter's genome; (13) Identified SVs (purple rectangles) in this region; (14) SV entries in the DGV database; (15) SNP-taggable SVs (McCarroll, et al., 2008); (16) Phred-like consensus scores; (17) RMS scores; (18) Read depth.

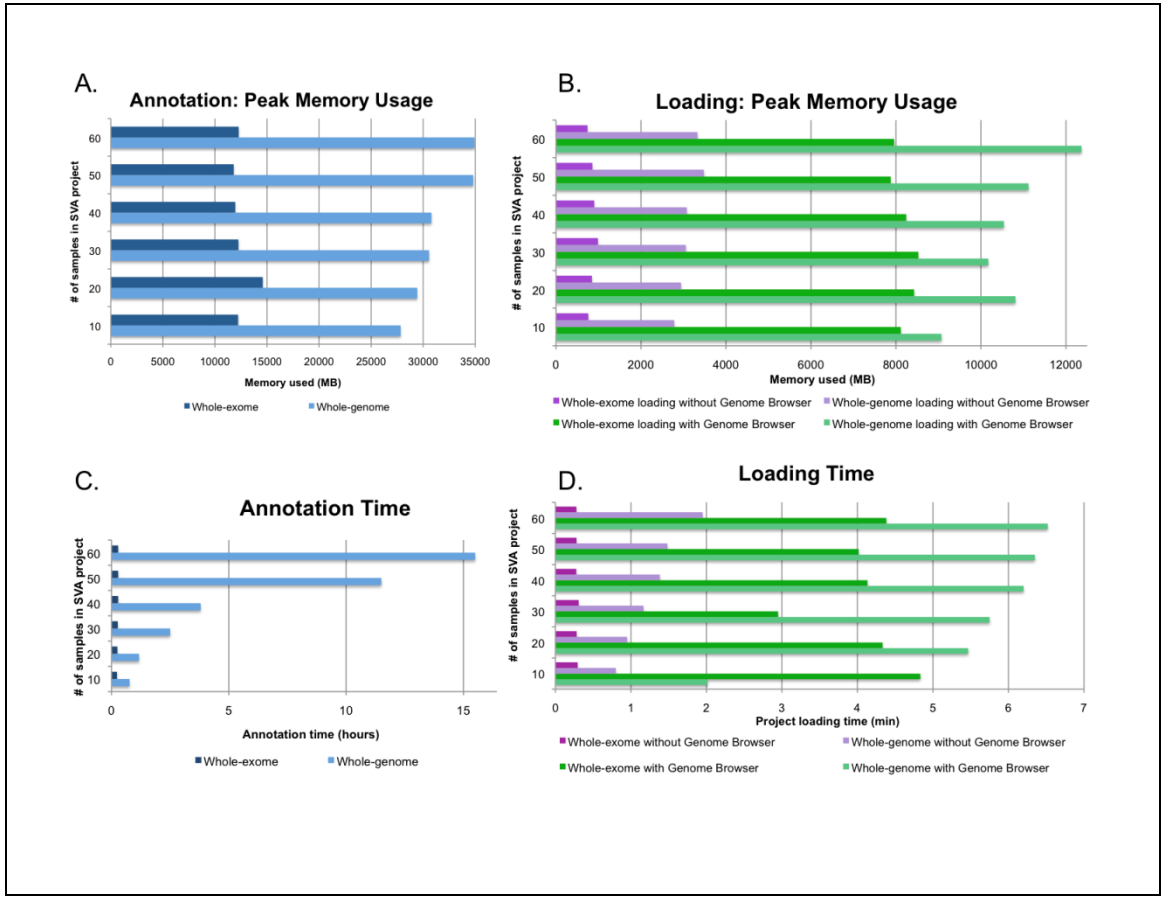


Figure S6. SVA's computational performance on a Linux workstation. Each performance test was done using SVA projects with varying numbers of genomes or exomes (10-60). The top two graphs show the peak memory usage during project annotation (A), project loading with and without genome browser (B). The bottom two graphs show the time requirements for annotating a project (C) and loading an annotated project (with or without initializing the Genome Browser) (D). This data is reflective of the computer used and other factors (e.g., CPU load) at the time of each performance test.

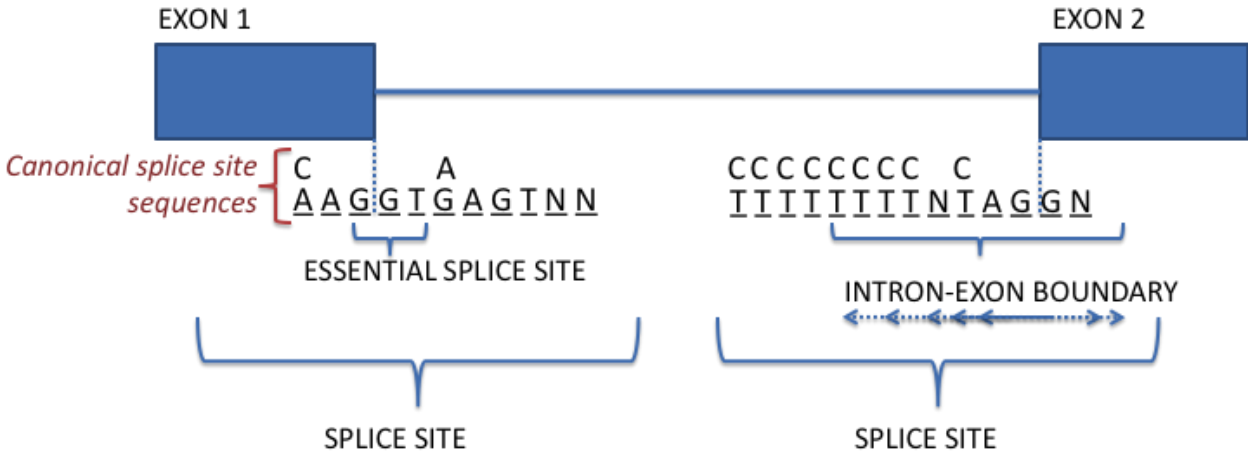


Figure S7. A diagram of the functional categories related to splicing sites: intron-exon boundary, essential splice site, and splice site. SVA annotates any variant located 1 bp into an exon and 2 bp into the intron at the exon-intron boundary as an "essential splice-site variant." It also lists any variants within the "intron-exon boundary" as defined by the user in the project (.gsap) file. The intron-exon boundary region in the above example (and the default setting in SVA) is 8 bps into the intron and 3 bps into the exon. In addition, SVA searches the variants to specifically identify mutations that disrupt conserved splicing sequences. As a first step, SVA searches the reference sequence for any of the conserved splice site sequences at the boundaries of the two exons annotated to interface following splicing (the conserved sequence must be present at both exon-intron boundaries). The canonical sequences screened by SVA include any combination of the bases specified in the figure at the designated positions. For example, if the program identifies the sequence CAGGTGAGTNN, beginning 3 bp into the exon, at the Exon 1-intron 1 boundary, and TTTTCCCCNTAGGGN at the Intron 1-Exon 2 boundary, this would be flagged as a canonical splice site. If flagged as a canonical splice site, SVA will annotate any variants that mutate this conserved sequence to a non-conserved sequence. These mutations that specifically disrupt conserved splice site sequences are assigned to the "splice-site" functional category.

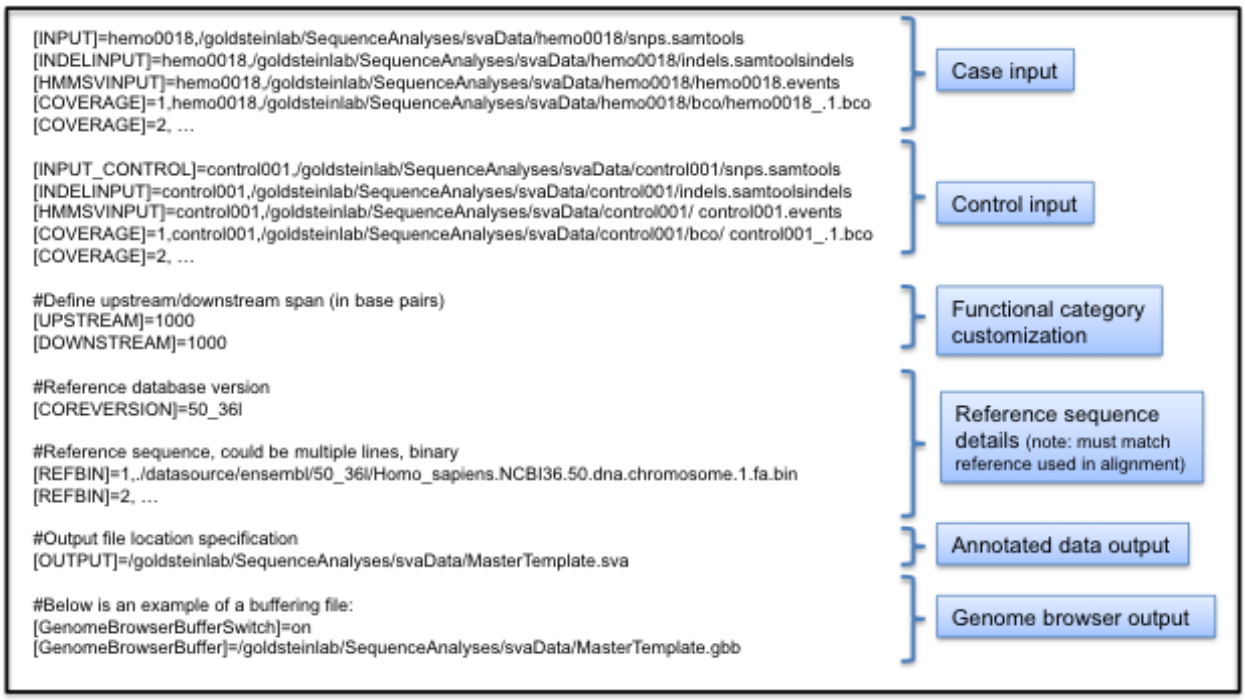


Figure S8. *Displayed is a subset of entries in the SVA project script file (.gsap).*

Table S1. The functional classifications used to categorize variants in SVA.

Functional category	Definition used in SVA	SNV	INDEL	SV
stop gained	a variant introducing a premature TAG, TAA, or TGA stop codon in a transcript of a protein-coding gene	x		
stop lost	a variant causing the loss of a TAG, TAA, or TGA stop codon in a transcript of a protein-coding gene	x		
non-synonymous coding	a variant located in a codon of a protein-coding gene resulting in a change from one amino acid residue to another, excluding variants that can be defined as the above two categories	x		
synonymous coding	a variant located in a codon of a protein-coding gene but not resulting in a change from one amino acid residue to another	x		
splice site	a variant that disrupts conserved canonical splicing sequences. *Note: this will frequently overlap with essential splice site. See Figure S7.*	x	x	
essential splice site	a variant changing the highly conserved GU in the first two bases of the intron or the AG in the last two bases of the intron. *See Figure S7.*	x		
intron-exon boundary	a variant occurring in any of the first *N1* nucleotides into the intron, or *N2* nucleotides into the exon . *N1* by default is 8 and *N2* by default is 3. The user can define *N1* and *N2* in the project file (*.gsap*). *Note: if a variant can be defined as "essential splice site", it will not be included in this category again. See Figure S7.*	x	x	x
5' UTR	a variant located within the 5' UTR of a transcript of a protein-coding gene	x	x	x
3' UTR	a variant located within the 3' UTR of a transcript of a protein-coding gene	x	x	x
exonic non-coding RNA	a variant occurring in an exon of a non-coding RNA	x		
upstream	an intergenic variant occurring within *N* nucleotides from the transcript start site of a protein-coding gene. *N* by default is 1000, and can be defined by the user	x	x	x
downstream	an intergenic variant occurring within *N* nucleotides from the transcript end site of a protein-coding gene. *N* by default is 1000, and can be defined by the user	x	x	x
intronic	a variant in the intron of a of a protein-coding gene, excluding those classified as a splice site or an intron-exon boundary	x	x	x
intergenic	a variant not located within a protein-coding gene, excluding those classified as upstream or downstream variants	x	x	x
coding disrupting frameshift	an indel that is located in the protein-coding portion of a gene and that is not a multiple of three, and thus will cause a frameshift in the resulting protein (which often also involves stop codon changes followed by a number of incorrect amino		x	

	acids, but may or may not immediately introduce or remove a stop codon)		
coding disrupting	an indel that is: (1) located in the protein-coding portion of a gene and is a multiple of three, and thus will cause coding changes (including the introduction or removal of a stop codon) but will not cause a frameshift in the resulting protein or (2) located in an RNA that is not predicted to be protein-coding (despite the functional category name suggesting otherwise)	x	
transcript included	used to indicate that an indel or structural variant occurs in a location that encompasses the full length of a protein-coding gene transcript.	x	x
coding disrupted	a structural variant that overlaps part of the coding sequence of a known protein-coding gene, but does not cover the whole length of any transcript of that gene		x

The status of protein-coding or non-coding RNA are determined by the annotation given in Ensembl release 50_36l (build 36, June 2008 version) by default. The use of the phrase "protein-coding gene" indicates that this region is annotated as a transcribed and translated gene in Ensembl release 50 (or build 36, June 2008 version) by default. Some functional categories are specific to only a single type of variant; an 'x' in the last three columns indicates which types of categories apply to SNVs, indels, and SVs.

References

Ashburner, M., *et al.* (2000) Gene ontology: tool for the unification of biology. The Gene Ontology Consortium, *Nature Genetics*, **25**, 25-29.

Bentley, D.R., *et al.* (2008) Accurate whole human genome sequencing using reversible terminator chemistry, *Nature*, **456**, 53-59.

Do, C.B., *et al.* (2005) ProbCons: Probabilistic consistency-based multiple sequence alignment, *Genome Research*, **15**, 330-340.

Durbin, R.M., *et al.* (2010) A map of human genome variation from population-scale sequencing, *Nature*, **467**, 1061-1073.

Flicek, P., *et al.* (2010) Ensembl's 10th year, *Nucleic Acids Research*, **38**, D557-562.

Frazer, K.A., *et al.* (2007) A second generation human haplotype map of over 3.1 million SNPs, *Nature*, **449**, 851-861.

Friedman, N., *et al.* (2002) A structural EM algorithm for phylogenetic inference, *Journal of Computational Biology*, **9**, 331-353.

http://www.1000genomes.org/node/101 (2011) VCF Format.

http://www.illumina.com/products/human1m_duo_dna_analysis_beadchip_kits.ilmn Illumina Infinium HD Human1M BeadChip: a commercial GWAS (genome-wide association studies) genotyping chip manufactured by Illumina Inc. .

http://www.ncbi.nlm.nih.gov/omim/ (2009) Online Mendelian Inheritance in Man, OMIM. McKusick-Nathans Institute of Genetic Medicine, Johns Hopkins University (Baltimore, MD) and National Center for Biotechnology Information, National Library of Medicine (Bethesda, MD)

Iafrate, A.J., *et al.* (2004) Detection of large-scale variation in the human genome, *Nature Genetics*, **36**, 949-951.

Kanehisa, M., *et al.* (2010) KEGG for representation and analysis of molecular networks involving diseases and drugs, *Nucleic Acids Research*, **38**, D355-360.

Levy, S., *et al.* (2007) The diploid genome sequence of an individual human, *PLoS Biology*, **5**, 254.

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

McCarroll, S., *et al.* (2008) Integrated detection and population-genetic analysis of SNPs and copy number variation, *Nature Genetics*, **40**, 1166 - 1174.

Pruitt, K.D., Tatusova, T. and Maglott, D.R. (2007) NCBI reference sequences (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins, *Nucleic Acids Research*, **35**, D61-65.

Sherry, S.T., *et al.* (2001) dbSNP: the NCBI database of genetic variation, *Nucleic Acids Research*, **29**, 308-311.

Smit, A.F., Hubley, R. and Green, P. (2010) RepeatMasker. *http://repeatmasker.org*.

Stone, E.A. and Sidow, A. (2005) Physicochemical constraint violation by missense substitutions mediates impairment of protein function and disease severity, *Genome Research*, **15**, 978-986.

Zhu, M., *et al.* (*Manuscript in progress*) Detection of copy number variation using whole genome sequence data from one hundred human genomes.