SVanalyzer Documentation

Release 0.12

Nancy F. Hansen

Oct 27, 2020

Contents

CHAPTER 1

SVbenchmark

SVbenchmark compares a set of "test" structural variants in VCF format to a known truth set (also in VCF format) and outputs estimates of sensitivity and specificity.

1.1 Usage

```
svanalyzer benchmark --ref <reference FASTA file> --test <VCF-formatted file of␣
↪variants to test> --truth <VCF-formatted file of true variants>
```

1.2 Options

Option	Description
–help|–manua	Display documentation.
–ref	The reference FASTA file for the supplied VCF file or files (required).
–test	A VCF-formatted file of structural variants to test (required).
–truth	A VCF-formatted file of variants to compare against (required).
–maxdist	Disallow matches if positions of two variants are more than maxdist bases from each other (default 100,000).
–normshift	Disallow matches if alignments between alternate alleles have normalized shift greater than normshift (default 1.0)
–norm-sizediff	Disallow matches if alternate alleles have normalized size difference greater than normsizediff (default 1.0)
–normdist	Disallow matches if alternate alleles have normalized edit distance greater than normdist (default 1.0)
–minsize	Only include true variants of size >= minsize for recall calculation and test variants >= minsize for precision calculation (default 0)
–prefix	Prefix for output file names (default: "benchmark")

1.3 Description

For sequence-specified test and truth structural variants in VCF files (i.e., files with ATGC sequences in the REF and ALT fields), SVbenchmark aligns constructed alternate haplotypes of each test/truth variant pair separated by no more than the distance specified by the –maxdist option to determine if the pair represent two equivalent variants.

In the false positive output VCF file, the program reports all test variants that are not equivalent to any true variant. In the false negative output VCF file, the program reports all true variants that are not equivalent to any test variant. The recall rate is reported in the report file as the percentage of true variants that are not false negatives, and the precision is reported as the percentage of test variants that are not false positives.

As of SVanalyzer v0.33, SVbenchmark will include non-sequence-specified deletions in its comparisons so long as the ALT field values of the VCF deletion records are "<DEL>" and an END value is include in the INFO field (e.g., END=5289355).

CHAPTER 2

SVmerge

SVmerge groups structural variants from a VCF file by calculating a distance matrix, then finding connected components of a graph in which the nodes are the variants and edges exist when the distances are below the specified maximum values.

The program steps through a set of structural variants, calculating distances to other nearby variants by comparing their alternate haplotypes. The program then reports clusters of variants, and prints a VCF file of "unique" variants, where the variant reported in the VCF record is a randomly-chosen representative from the largest cluster (or a randomly selected largest cluster, in the case of a tie among cluster sizes) of exactly matching variants.

Alternatively, a file of previously-calculated distances can be provided with the –distance_file option, and the clustering can be skipped with the option –skip_clusters.

NOTE: SVmerge only clusters and merges sequence-specific variants, i.e., structural variants with ATGCN sequences for their REF and ALT alleles, or deletions with a valid "END" INFO tag. These variants will be printed as singletons unless the –seqspecific option is specified (see below).

2.1 Usage

```
svanalyzer merge --ref <reference FASTA file> --variants <VCF-formatted variant file>␣
↪--prefix <prefix for output files>
svanalyzer merge --ref <reference FASTA file> --fof <file of paths to VCF-formatted␣
↪variant files> --prefix <prefix for output files>
```

2.2 Options

Option	Description
–help|–manual	Display documentation.
–ref	The reference FASTA file for the supplied VCF file or files.
–variants	A VCF-formatted file of (possibly equivalent) variants to merge.
–fof	A file of paths to VCF-formatted files to merge.
–prefix	Prefix for output file names (default "merged")
–maxdist	Maximum distance between pairs of variants to perform comparison for potential merging (default: 2000)
–reldist	Maximum allowable edit distance, normalized by the mean length of larger allele for the two variants, in an alignment used to merge two variants (default: 0.2)
–relsizediff	Maximum allowable alt allele size difference, normalized by the mean length of larger allele for the two variants, to merge two variants (default: 0.2)
–relshift	Maximum allowable shift, normalized by the mean length of the larger allele for the two variants, in an alignment used to merge two variants (default: 0.2)
–seqspecific	With this option, SVmerge will fail to print out any SV that does not have an ATGCN sequence for REF and ALT in the input VCF files.

CHAPTER 3

SVcomp

SVcomp calculates "distances" between pairs of structural variants in VCF format by constructing their alternate haplotypes and aligning them to each other.

3.1 Usage

```
svanalyzer comp --ref reference.fasta --first <first VCF-formatted file> --second
↪<second VCF-formatted file>
```

3.2 Options

Option	Description
–help|–manual	Display documentation.
–ref	The reference FASTA file for the supplied VCF file or files.
–first	A VCF-formatted file of variants to compare
–second	Second VCF-formatted file of variants to compare–must have the same number of variants as the first file

CHAPTER 4

SVwiden

SVwiden reads a VCF file and uses MUMmer to determine widened coordinates for structural variants, adding custom tags to the VCF record.

4.1 Usage

```
svanalyzer widen --ref <reference FASTA file> --variants <VCF-formatted variant file>␣
↪--prefix <prefix for output files>
```

4.2 Options

Option	Description
–help|–manual	Display documentation.
–ref	The reference FASTA file for the supplied VCF file or files.
–variants	A VCF-formatted file of (possibly equivalent) variants to merge.
–fof	A file of paths to VCF-formatted files to merge.
–prefix	Prefix for output file names (default: “widened”)

CHAPTER 5

SVrefine

SVrefine reads a delta-formatted file of MUMmer alignments of an assembly to the reference to call structural variants (or refine variants in chosen genomic regions) and print them out in VCF format.

5.1 Usage

```
SVrefine.pl --delta <path to delta file of alignments> --regions <path to BED-
↪formatted file of regions> --ref_fasta <path to reference multi-FASTA file> --query_
↪fasta <path to query multi-FASTA file> --outvcf <path to output VCF file> --
↪svregions <path to output BED file of SV regions> --outref <path to bed file of␣
↪homozygous reference regions> --nocov <path to bed file of regions with no coverage>
```

5.2 Options

Option	Description
–help|–manual	Display documentation.
–delta	Path to a delta file produced by MUMmer with alignments to be used for retrieving SVs.
–regions	Path to a BED file of regions to be investigated for structural variants in the assembly (Optional).
–ref_fasta	Path to a multi-fasta file containing the sequences used as a reference in the MUMmer alignment (Optional).
–query_fasta	Path to a multi-fasta file containing the sequences used as a query in the MUMmer alignment (Optional).
–outvcf	Path to which to write a new VCF-formatted file of structural variants.
–refname	String to include as the reference name in the VCF header.
–sample-name	String to include as the sample name in the output VCF file.
–maxsize	Specify an integer for the maximum size of SV to report.
–noheader	Flag option to suppress printout of the VCF header.
–nocov	Path to write a BED file with "no coverage" regions (only used when –regions option is specified).

SVanalyzer is a software package for the analysis of large insertions, deletions, and inversions in DNA. SVanalyzer tools use repeat-aware methods to refine, compare, and cluster different structural variant calls.

CHAPTER 6

Install

6.1 Using conda

SVanalyzer can be installed using the conda package manager with the bioconda channel. For details on setting up conda/bioconda, see the Bioconda user docs.

```
conda create -n svanalyzer
conda activate svanalyzer
conda install svanalyzer
```

6.2 With a release tarball/github clone

SVanalyzer can also be installed by downloading a release tarball or cloning the github repository:

```
git clone https://github.com/nhansen/SVanalyzer.git
```

After unzipping the tarball or cloning the directory, build SVanalyzer:

```
cd SVanalyzer
perl Build.PL
./Build
./Build test
./Build install
```

To install SVanalyzer to an alternate location (e.g., if you do not have root permissions), call "perl Build.PL –install_base $HOME".

CHAPTER 7

Command documentation

- *SVbenchmark* - Compare a set of "test" structural variants in VCF format to a known truth set and report sensitivity and specificity
- *SVmerge* - Merge similar sequence-resolved SVs in VCF format
- *SVcomp* - Compare sequence-resolved SVs to each other
- *SVwiden* - Add tags to a VCF file of sequence-resolved SVs detailing surrounding repetitive genomic context
- *SVrefine* - Call sequence-resolved structural variants (SVs) from assembly consensus