# Package 'sveval'

June 14, 2019

**Title** SV evaluation

**Version** 1.2.1

**Description** Evaluate SV in a call set against a truth set using overlap-based approaches and sequence comparison for insertions.

**Depends** R (>= 3.4.4)

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.1.1

**Imports** VariantAnnotation,
GenomicRanges,
IRanges,
magrittr,
dplyr,
rlang,
DelayedArray,
Biostrings,
parallel,
testthat,
ggplot2,
shiny,
DiagrammeR,
DT

## R topics documented:

sveval-package *SV evaluation*

## Description

Evaluate SV in a call set against a truth set using overlap-based approaches and sequence comparison for insertions.

## Details

| | |
|---|---|
| Package: | sveval |
| Type: | Package |
| Version: | 1.2.1 |
| Date: | 2019-02-28 |
| License: | MIT |

## Author(s)

Jean Monlong <jmonlong@ucsc.edu>

## See Also

http://www.github.com/jmonlong/sveval

## Examples

```
## Not run:
eval = svevalOl('calls.vcf', 'truth.vcf')
plot_prcurve(eval$curve)

# Comparing multiple methods
eval.1 = svevalOl('calls1.vcf', 'truth.vcf')
eval.2 = svevalOl('calls2.vcf', 'truth.vcf')
plot_prcurve(list(eval.1$curve, eval.2$curve), labels=c('method1', 'method2'))

## End(Not run)
```

## `filterSVs` *Filter SVs for size and regions of interest*

### Description

Filter SVs for size and regions of interest

### Usage

```
filterSVs(sv.gr, regions.gr = NULL, ol.prop = 0.5, min.size = 0,
  max.size = Inf)
```

### Arguments

| | |
|---|---|
| `sv.gr` | the input SVs (e.g. read from `readSVvcf`) |
| `regions.gr` | the regions of interest. Ignored if NULL (default). |
| `ol.prop` | minimum proportion of sv.gr that must overlap regions.gr. Default is 0.5 |
| `min.size` | the minimum SV size to be considered. Default 0. |
| `max.size` | the maximum SV size to be considered. Default is Inf. |

### Value

a subset of sv.gr that overlaps regions.gr or in the specified size range.

### Author(s)

Jean Monlong

## `findNocalls` *Find no-calls variants*

### Description

Compare calls with a truth set and identifies which variants from the truth set specifically not called (genotype ./.).

### Usage

```
findNocalls(calls.gr, truth.gr, max.ins.dist = 20, min.cov = 0.5,
  min.del.rol = 0.1, ins.seq.comp = FALSE, nb.cores = 1,
  sample.name = NULL, check.inv = FALSE)
```

## Arguments

| | |
|---|---|
| `calls.gr` | call set. A GRanges or the path to a VCF file. |
| `truth.gr` | truth set. A GRanges or the path to a VCF file. |
| `max.ins.dist` | maximum distance for insertions to be clustered. Default is 20. |
| `min.cov` | the minimum coverage to be considered a match. Default is 0.5 |
| `min.del.rol` | minimum reciprocal overlap for deletions. Default is 0.1 |
| `ins.seq.comp` | compare sequence instead of insertion sizes. Default is FALSE. |
| `nb.cores` | number of processors to use. Default is 1. |
| `sample.name` | the name of the sample to use if VCF files given as input. If NULL (default), use first sample. |
| `check.inv` | should the sequence of MNV be compared to identify inversions. |

## Details

Same overlapping strategy as in `sveval01` although here no-calls are kept and there is no splitting by genotype.

## Value

a data.frame with coordinates and variant ids from the truth set corresponding to no-calls.

## Author(s)

Jean Monlong

---

`ivg_sv` *Interactive exploration of SVs in a variation graph*

---

## Description

Opens a Shiny app with a dynamic table that contains input SVs. Clicking on a SV (row in the table) generates a simplified representation of the variation graph around this SV. The number of flanking nodes (context) can be increased if necessary, e.g. for large insertions. vg needs to be installed (https://github.com/vgteam/vg).

## Usage

```
ivg_sv(svs, xg, ucsc.genome = "hg38")
```

## Arguments

| | |
|---|---|
| `svs` | either a GRanges with SVs (e.g. from `readSVvcf`) or the path to a VCF file. |
| `xg` | the path to the xg object of the variation graph. |
| `ucsc.genome` | the genome version for the UCSC Genome Browser automated link. |

### Value

Starts a Shniy app in a web browser.

### Author(s)

Jean Monlong

---

## `plot_prcurve` *Create precision-recall graphs*

---

### Description

Create a precision/recall curve using metrics computed by the `sveval01` function. The `sveval01` function returns a list containing a "curve" data.frame with the evaluation metrics for different quality thresholds.

### Usage

```
plot_prcurve(eval, labels = NULL)
```

### Arguments

| | |
|---|---|
| `eval` | a data.frame, a list of data.frames, or a vector with one or several paths to files with "curve" information. |
| `labels` | the labels to use for each input (when multiple inputs are used). Ignored is NULL (default). |

### Details

If the input is a data.frame (or list of data.frames) it should be the "curve" element of the list returned by the `sveval01` function. If the input is a character (or a vector of characters), they are considered to be file names and the data will be read from these files.

If multiple inputs are given, either using a list of data.frames or a vectors with several filenames, one curve per input will be created. This is to be used to quickly compare several methods. The "labels" parameters can be used to specify a label for each input to use for the graphs.

### Value

list of ggplot graph objects

### Author(s)

Jean Monlong

## Examples

```
## Not run:
eval = svevalOl('calls.vcf', 'truth.vcf')
plot_prcurve(eval$curve)

# Comparing multiple methods
eval.1 = svevalOl('calls1.vcf', 'truth.vcf')
eval.2 = svevalOl('calls2.vcf', 'truth.vcf')
plot_prcurve(list(eval.1$curve, eval.2$curve), labels=c('method1', 'method2'))

# Or if the results were previously written in files
plot_prcurve(c('methods1-prcurve.tsv', 'methods2-prcurve.tsv'), labels=c('method1', 'method2'))

## End(Not run)
```

---

## `readSVvcf` *Read SVs from a VCF file*

---

### Description

Read a VCF file that contains SVs and create a GRanges with relevant information, e.g. SV size or genotype quality.

### Usage

```
readSVvcf(vcf.file, keep.ins.seq = FALSE, sample.name = NULL,
  qual.field = c("GQ", "QUAL"), check.inv = FALSE, keep.ids = FALSE,
  nocalls = FALSE, right.trim = TRUE)
```

### Arguments

| | |
|---|---|
| `vcf.file` | the path to the VCF file |
| `keep.ins.seq` | should it keep the inserted sequence? Default is FALSE. |
| `sample.name` | the name of the sample to use. If NULL (default), use first sample. |
| `qual.field` | fields to use as quality. Will be tried in order. |
| `check.inv` | should the sequence of MNV be compared to identify inversions. |
| `keep.ids` | keep variant ids? Default is FALSE. |
| `nocalls` | if TRUE returns no-calls only (genotype ./.). Default FALSE. |
| `right.trim` | if TRUE (default) the REF/ALT sequences are right-trimmed after splitting up multi-ALT variants. |

### Details

By default, the quality information is taken from the QUAL field. If all values are NA or 0, the function will try other fields as speficied in the "qual.field" vector. Fields can be from the INFO or FORMAT fields.

## Value

a GRanges object with relevant information.

## Author(s)

Jean Monlong

## Examples

```
## Not run:
calls.gr = readSVvcf('calls.vcf')

## End(Not run)
```

---

`svevalOl` *SV evaluation based on overlap and variant size*

---

## Description

SV evaluation based on overlap and variant size

## Usage

```
svevalOl(calls.gr, truth.gr, max.ins.dist = 20, min.cov = 0.5,
  min.del.rol = 0.1, ins.seq.comp = FALSE, nb.cores = 1,
  min.size = 50, max.size = Inf, bed.regions = NULL,
  bed.regions.ol = 0.5, qual.field = c("QUAL", "GQ"),
  sample.name = NULL, outfile = NULL, out.bed.prefix = NULL,
  qual.ths = c(0, 2, 3, 4, 5, 7, 10, 12, 14, 21, 27, 35, 45, 50, 60, 75,
  90, 99, 110, 133, 167, 180, 250, 350, 450, 550, 650),
  qual.quantiles = seq(0, 1, 0.1), check.inv = FALSE,
  geno.eval = FALSE, stitch.hets = FALSE, stitch.dist = 20,
  merge.hets = FALSE, merge.rol = 0.8)
```

## Arguments

| | |
|---|---|
| `calls.gr` | call set. A GRanges or the path to a VCF file. |
| `truth.gr` | truth set. A GRanges or the path to a VCF file. |
| `max.ins.dist` | maximum distance for insertions to be clustered. Default is 20. |
| `min.cov` | the minimum coverage to be considered a match. Default is 0.5 |
| `min.del.rol` | minimum reciprocal overlap for deletions. Default is 0.1 |
| `ins.seq.comp` | compare sequence instead of insertion sizes. Default is FALSE. |
| `nb.cores` | number of processors to use. Default is 1. |
| `min.size` | the minimum SV size to be considered. Default 50. |
| `max.size` | the maximum SV size to be considered. Default is Inf. |

| | |
|---|---|
| `bed.regions` | If non-NULL, a GRanges object or path to a BED file (no headers) with regions of interest. |
| `bed.regions.ol` | minimum proportion of sv.gr that must overlap regions.gr. Default is 0.5 |
| `qual.field` | fields to use as quality. Will be tried in order. |
| `sample.name` | the name of the sample to use if VCF files given as input. If NULL (default), use first sample. |
| `outfile` | the TSV file to output the results. If NULL (default), returns a data.frame. |
| `out.bed.prefix` | prefix for the output BED files. If NULL (default), no BED output. |
| `qual.ths` | the QUAL thresholds for the PR curve. If NULL, will use quantiles. see `qual.quantiles`. |
| `qual.quantiles` | the QUAL quantiles for the PR curve, if qual.ths is NULL. Default is (0, .1, ..., .9, 1). |
| `check.inv` | should the sequence of MNV be compared to identify inversions. |
| `geno.eval` | should het/hom be evaluated separately (genotype evaluation). Default FALSE. |
| `stitch.hets` | should clustered hets be stitched together before genotype evatuation. Default is FALSE. |
| `stitch.dist` | the maximum distance to stitch hets during genotype evaluation. |
| `merge.hets` | should similar hets be merged into homs before genotype evaluation. Default is FALSE. |
| `merge.rol` | the minimum reciprocal overlap to merge hets before genotype evaluation. |

## Value

a list with

| | |
|---|---|
| `eval` | a data.frame with TP, FP and FN for each SV type when including all variants |
| `curve` | a data.frame with TP, FP and FN for each SV type when using different quality thesholds |
| `svs` | a list of GRanges object with FP, TP and FN for each SV type (when using QUAL>=0 threshold). |

## Author(s)

Jean Monlong

## Examples

```
## Not run:
## From VCF files
eval = svevalOl('calls.vcf', 'truth.vcf')

## From GRanges
calls.gr = readSVvcf('calls.vcf')
truth.gr = readSVvcf('truth.vcf')
eval = svevalOl(calls.gr, truth.gr)
```

```
## Genotype evaluation
eval = svevalOl(calls.gr, truth.gr, geno.eval=TRUE, merge.hets=TRUE, stitch.hets=TRUE)

## End(Not run)
```

## svOverlap *Overlap and annotate SV sets with coverage metrics*

### Description

Overlap and annotate SV sets with coverage metrics

### Usage

```
svOverlap(query, subject, max.ins.dist = 20, min.cov = 0.5,
  min.del.rol = 0.1, ins.seq.comp = FALSE, nb.cores = 1)
```

### Arguments

| | |
|---|---|
| `query` | a GRanges object with SVs |
| `subject` | another GRanges object with SVs |
| `max.ins.dist` | maximum distance for insertions to be clustered. Default is 20. |
| `min.cov` | the minimum coverage to be considered a match. Default is 0.5 |
| `min.del.rol` | minimum reciprocal overlap for deletions. Default is 0.1 |
| `ins.seq.comp` | compare sequence instead of insertion sizes. Default is FALSE. |
| `nb.cores` | number of processors to use. Default is 1. |

### Value

a list with:

| | |
|---|---|
| `query` | the query GRanges object annotated |
| `subject` | the subject GRanges object annotated |

### Author(s)

Jean Monlong

# Index