SvAnna Documentation

Release 1.0.4-SNAPSHOT

Daniel Danis, Peter N Robinson

Apr 12, 2023

CONTENTS:

Efficient and accurate pathogenicity prediction for coding and regulatory structural variants in long-read genome sequencing.

SvAnna performs phenotype-driven prioritization of structural variants in VCF files, focusing specifically on long-read WGS analysis of germline variants.

CHAPTER

ONE

QUICKSTART

This document is intended for the impatient users who want to quickly setup and prioritize variants with SvAnna.

1.1 Prerequisites

SvAnna is written in Java 11 and needs Java 11+ to be present in the runtime environment. Please verify that you are using Java 11+ by running:

```
$ java -version
```

If `java` is present on your `$PATH`, then the command above will print a message similar to this one:

```
openjdk version "11" 2018-09-25
OpenJDK Runtime Environment 18.9 (build 11+28)
OpenJDK 64-Bit Server VM 18.9 (build 11+28, mixed mode)
```

1.2 Setup

SvAnna is installed by running the following three steps.

1.2.1 1. Download SvAnna distribution ZIP

Download and extract SvAnna distribution ZIP archive from here. Expand the *Assets* menu and download the `svanna-cli-${project.version}-distribution.zip`. Choose the latest stable version, or a release candidate (RC).

After unzipping the distribution archive, run the following command to display the help message:

```
$ java -jar svanna-cli-${project.version}.jar --help
```

Note: If things went OK, the command above will print the following help message:

```
Structural variant prioritization
Usage: svanna-cli.jar [-hV] [COMMAND]
  -h, --help      Show this help message and exit.
  -V, --version   Print version information and exit.
Commands:
```

(continues on next page)

(continued from previous page)

```
  prioritize        Prioritize the variants.
See the full documentation at `https://svanna.readthedocs.io/en/master`
```

1.2.2 2. Download SvAnna database files

SvAnna database files are available for download in the rstdownloads section.

After the download, unzip the archive(s) content into a folder of your choice and note down the path:

```
$ unzip -d svanna-data *.svanna.zip
```

1.3 Prioritize structural variants in VCF file

Let's annotate a toy VCF file containing eight SVs reported in the SvAnna manuscript. First, let's download the VCF file from here:

```
$ wget https://raw.githubusercontent.com/TheJacksonLaboratory/SvAnna/master/svanna-cli/
↪src/examples/example.vcf
```

The variants were sourced from published clinical case reports and presence of each variant results in a Mendelian disease.

For the purpose of this test run, let's assume that the VCF file contains SVs identified in a short/long read sequencing run of a patient presenting with the following clinical symptoms:

- *HP:0011890* - Prolonged bleeding following procedure
- *HP:0000978* - Bruising susceptibility
- *HP:0012147* - Reduced quantity of Von Willebrand factor

Now, let's prioritize the variants:

```
$ java -jar svanna-cli-${project.version}.jar prioritize -d svanna-data --output-format␣
↪html,csv,vcf --vcf example.vcf --phenotype-term HP:0011890 --phenotype-term HP:0000978␣
↪--phenotype-term HP:0012147
```

The variant `Othman-2010-20696945-VWF-index-FigS7` disrupts a promoter of the *von Willenbrand factor* (*VWF*) gene (Othman et al., 2010). The variant receives the highest PSV score of 47.26, and it is ranked first.

SvAnna stores prioritization results in *HTML*, *CSV*, and *VCF* output formats in the current working directory.

CHAPTER

TWO

SETTING UP SVANNA

SvAnna is a desktop Java application that requires several external files to run. This document explains how to download the external files and how to prepare SvAnna for running in the local system.

Note: SvAnna is written with Java version 11 and will run and compile under Java 11+.

2.1 Installation

To install SvAnna, you need to get SvAnna distribution ZIP archive that contains the executable JAR file, and SvAnna database files.

2.1.1 Prebuilt SvAnna executable

To download the executable SvAnna JAR file, go to the Releases section on the SvAnna GitHub page and download the latest SvAnna ZIP archive.

2.1.2 SvAnna database files

SvAnna database files are available for download in the rstdownloads section.

After the download, unzip the archive and put SvAnna database files into a folder of your choice:

```
$ unzip -d svanna-data *.svanna.zip
```

Note: From now on, we will use `svanna-data` instead of spelling out the full path to SvAnna database files.

2.1.3 Build SvAnna from source

As an alternative to using prebuilt SvAnna JAR file, the SvAnna JAR file can also be built from Java sources.

SvAnna was written with Java version 11. Git and Java Development Kit version 11 or better are required for build.

Run the following commands to download SvAnna source code from GitHub repository and to build SvAnna JAR file:

```
$ git clone https://github.com/TheJacksonLaboratory/SvAnna
$ cd SvAnna
$ ./mvnw package
```

After the build, the JAR file is located at `svanna-cli/target/svanna-cli-${project.version}.jar`:

```
$ java -jar svanna-cli/target/svanna-cli-${project.version}.jar --help
```

Note: From now on, we will use `svanna-cli.jar` instead of spelling out the full path to the JAR file within your environment.

CHAPTER

THREE

RUN SVANNA

SvAnna is a command-line Java tool that runs with Java version 11 or higher.

In the examples below, we assume that `svanna-cli.jar` points to the executable JAR file and `svanna-data` points to the data directory we created in the *Setting up SvAnna* section.

3.1 Prioritization of structural variants

SvAnna provides `prioritize` command for performing phenotype-driven prioritization of structural variants (SVs) stored in VCF format. The prioritized variants are stored in one or more *Output formats*.

To prioritize variants in the example.vcf file (an example VCF file with 8 variants stored in SvAnna repository), run:

```
$ java -jar svanna-cli.jar prioritize -d svanna-data --vcf example.vcf --phenotype-term␣
↪HP:0011890 --phenotype-term HP:0000978 --phenotype-term HP:0012147 --out-dir results --
↪prefix example
```

After the run, the results are stored at `results/example.html`.

3.1.1 Mandatory arguments

All CLI arguments for the `prioritize` command are supplied as *options* (no positional parameters).

There is one *mandatory* option:

- `-d | --data-directory` - path to SvAnna data directory.

Analysis input

The input data can be specified in two ways: either as a path to a VCF file along with one or more HPO terms, or as a *phenopacket*:

- `-p | --phenopacket` - path to a phenopacket file. We support *v1* and *v2* schemas and the file can be in JSON, YAML, or protobuf binary format.
- `-t | --phenotype-term` - HPO term describing clinical condition of the proband, may be specified multiple times (e.g. `--term HP:1234567 --term HP:9876543`).
- `--vcf` - path to the input VCF file.

Note: In case path to a VCF file is provided both in *phenopacket* and via `--vcf` option, the `--vcf` option has a precedence.

3.1.2 Optional parameters

SvAnna allows to fine-tune the prioritization using a number of *optional* parameters. For clarity, we group the options into several groups:

Run options

- `--frequency-threshold` - threshold for labeling SVs in population variant databases *pv* as common. If query SV *v* overlaps with *pv* that has frequency above the threshold, then *v* is considered to be *common*. The value is provided as a percentage (default `1`).
- `--overlap-threshold` - threshold to determine if a SV matches a variant from the population variant databases. The value is provided as a percentage (default `80`).
- `--min-read-support` - minimum number of reads supporting the presence of the *alt* allele required to include a variant into the analysis (default *3* `).
- `--n-threads` - number of threads used to prioritize the SVs (default `2`).

Output options

- `--no-breakends` - do not report breakends/translocations in the HTML report (default: `false`).
- `--output-format` - comma separated list of output formats to use for writing the results (default `html`).

Note: See *Output formats* section for more details.

- `--out-dir` - path to a folder where to write the output files (default: current working directory).
- `--prefix` - prefix for output files (default: based on the input VCF name).
- `--report-top-variants` - include top *n* variants in the HTML report (default: `100`).

Note: Beware, the HTML report becomes rather large when including large number of variants.

- `--uncompressed-output` - the tabular and VCF output files are compressed by default. Use this flag if you want to disable compressing the output files (default: `false`).

SvAnna configuration

- `--term-similarity-measure` - phenotype term similarity measure, use one of `{RESNIK_SYMMETRIC, RESNIK_ASYMETRIC}` (default: RESNIK_SYMMETRIC).
- `--ic-mica-mode` - the mode for getting information content of the most informative common ancestors for terms t_1, and t_2. Use one of `{DATABASE, IN_MEMORY}` (default: `DATABASE`).
- `--promoter-length` - number of bases pre-pended to a transcript and evaluated as a promoter region (default: `2000`).
- `--promoter-fitness-gain` - set to `0.` to score the promoter variants as strictly as coding variants or to `1.` to completely disregard the promoter variants (default: `0.6`).
- `-v` - set logging output granularity. The option can be set multiple times (e.g. `-vv`) to increase logging output.

See the next section to learn more about the SvAnna *Output formats*, and the *Examples* section to see how SvAnna prioritizes various SV classes.

CHAPTER

FOUR

OUTPUT FORMATS

SvAnna supports storing results in 4 output formats: *HTML*, *VCF CSV*, and *TSV*. Use the `--output-format` option to select one or more of the desired output formats (e.g. `--output-format html,vcf`).

4.1 HTML output format

SvAnna creates an *HTML* file with the analysis summary and with variants sorted by the PSV score in descending order. By default, top 100 variants are included into the report. The number of the reported variants can be adjusted by the `--report-top-variants` option.

The report consists of several parts:

- *Analysis summary* - Details of HPO terms of the proband, paths of the input files, and the analysis parameters.
- *Variant counts* - Breakdown of the number of the variant types of the different categories.
- *Prioritized SVs* - Visualizations of the prioritized variants.

Note: Only the variants that passed all the filters are visualized in the *Prioritized SVs* section.

The `--no-breakends` option excludes breakends/translocations from the report.

4.2 VCF output format

When including `vcf` into the `--output-format` option, a VCF file with all input variants is created. The prioritization adds a novel *INFO* field to each variant:

- `PSV` - an *INFO* field containing PSV score for the variant.

Note:

- `--report-top-variants` option has no effect for the *VCF* output format.
- Add `--uncompressed-output` flag if you want to get uncompressed VCF file.

4.3 CSV/TSV output format

To write the prioritization results into a *CSV* (or *TSV*) file, use `csv` (`tsv`) in the `--output-format` option.

The results are written into a tabular file with the following columns:

- *contig* - name of the contig/chromosome (e.g. `1`, `2`, `X`).
- *start* - 0-based start coordinate (excluded) of the variant on positive strand.
- *end* - 0-based end coordinate (included) of the variant on positive strand.
- *id* - variant ID as it was present in the input VCF file.
- *vtype* - variant type, one of `{DEL, DUP, INV, INS, BND, CNV}`.
- *failed_filters* - the names of filters that the variant failed to pass. The names are separated by semicolon (`;`) * `filter` - the variant failed previous VCF filters - at least one filter flag is present in the variant VCF line, except for `PASS`. * `coverage` - the variant is supported by less reads than specified by `--min-read-support` option.
- *psv* - the PSV score value.

Table 1: Tabular output

contig	start	end	id	vtype	failed_filters	psv
11	31130456	31671718	abcd	DEL		109.75766900764305
18	46962113	46969912	efgh	DUP	filter;coverage	3.2
…	…	…	…	…	…	…

Note:

- `--report-top-variants` option has no effect for the *CSV* and *TSV* output formats.
- Add `--uncompressed-output` flag if you want to get uncompressed VCF file.

CHAPTER

FIVE

EXAMPLES

This section shows how SvAnna prioritizes various structural variant classes. The resulting HTML reports contain graphics that are reported in the supplement of SvAnna paper.

The examples work with variants stored in examples.vcf file. The VCF file is stored in SvAnna GitHub repository. Use the run_examples.sh script to generate HTML reports for all cases described below. Note that you must enter the paths to SvAnna JAR file, data directory, and the `examples.vcf` into the script before running.

5.1 Single exon deletion

A deletion of 6.93 kb (`chr17:31,150,798-31,157,725del`) affecting *NF1* that was assigned a *PSV* score of 124.98.

The deletion affects exon 2 of several *NF1* transcripts. Pathogenic variants in *NF1* are associated with neurofibromatosis type 1 (`OMIM:162200`).

The phenotypic features curated for the proband `UAB-1` were:

- `HP:0007565` Multiple cafe-au-lait spots
- `HP:0009732` Plexiform neurofibroma
- `HP:0009735` Spinal neurofibromas
- `HP:0009736` Tibial pseudarthrosis

Data were curated from a published case report in Decoding NF1 Intragenic Copy-Number Variations.

5.1.1 Command

```
$ java -jar svanna-cli.jar prioritize -d path/to/svanna-data --vcf example.vcf --term␣
↪HP:0007565 --term HP:0009732 --term HP:0009735 --term HP:0009736
```

5.2 Deletion of multiple exons

A deletion of 10.26 kb (`chr17:43,100,079-43,110,335del`) affecting *BRCA1* that was assigned a *PSV* score of 272.91.

The deletion affects three *BRCA1* exons. Pathogenic variants in *BRCA1* are associated with Breast-ovarian cancer, familial, 1 (`OMIM:604370`).

The phenotypic feature curated for this case was:

- `HP:0003002` Breast carcinoma

Data were curated from a published case report The first case report of a large deletion of the BRCA1 gene in Croatia.

5.2.1 Command

```
$ java -jar svanna-cli.jar prioritize  -d path/to/svanna-data --vcf example.vcf --term␣
↪HP:0003002
```

5.3 Deletion of multiple genes

Deletion of 481.73 kb (`chr2:109,923,337-110,405,062del`) affecting *MALL*, *NPHP1*, and *MTLN* that was assigned a *PSV* score of 16.41.

Pathogenic variants in *NPHP1* are associated with Joubert syndrome 4 (`OMIM:609583`).

The phenotypic features curated for this case were:

- `HP:0003774` Stage 5 chronic kidney disease
- `HP:0001320` Cerebellar vermis hypoplasia
- `HP:0002078` Truncal ataxia
- `HP:0000618` Blindness
- `HP:0000508` Ptosis
- `HP:0002419` Molar tooth sign on MRI
- `HP:0011933` Elongated superior cerebellar peduncle
- `HP:0002070` Limb ataxia
- `HP:0000543` Optic disc pallor
- `HP:0000589` Coloboma

Data were curated from a published case report Whole-exome sequencing and digital PCR identified a novel compound heterozygous mutation in the NPHP1 gene in a case of Joubert syndrome and related disorders.

5.3.1 Command

```
$ java -jar svanna-cli.jar prioritize -d path/to/svanna-data --vcf example.vcf --term␣
↪HP:0003774 --term HP:0001320 --term HP:0002078 --term HP:0000618 --term HP:0000508 --
↪term HP:0002419 --term HP:0011933 --term HP:0002070 --term HP:0000543 --term HP:0000589
```

5.4 Duplication of coding sequence

Duplication of 36 bp (`chr13:72835296-72835332dup`) affecting *PIBF1* that was assigned a *PSV* score of 3.29. Pathogenic variants in *PIBF1* are associated with Joubert syndrome 33 (`OMIM:617767`).

The phenotypic features curated for this case were:

- `HP:0032417` Periglomerular fibrosis
- `HP:0000076` Vesicoureteral reflux

- HP:0002079 Hypoplasia of the corpus callosum
- HP:0001541 Ascites
- HP:0000540 Hypermetropia
- HP:0011968 Feeding difficulties
- HP:0001250 Seizure
- HP:0000490 Deeply set eye
- HP:0001263 Global developmental delay
- HP:0001284 Areflexia
- HP:0002240 Hepatomegaly
- HP:0001290 Generalized hypotonia
- HP:0031200 Hyaline casts
- HP:0011800 Midface retrusion
- HP:0000090 Nephronophthisis
- HP:0000092 Renal tubular atrophy
- HP:0001919 Acute kidney injury
- HP:0012650 Perisylvian polymicrogyria
- HP:0002419 Molar tooth sign on MRI
- HP:0002119 Ventriculomegaly
- HP:0000105 Enlarged kidney

Data were curated from a published case report A biallelic 36-bp insertion in PIBF1 is associated with Joubert syndrome

5.4.1 Command

```
$ java -jar svanna-cli.jar prioritize -d path/to/svanna-data --vcf example.vcf --term␣
↪HP:0032417 --term HP:0000076 --term HP:0002079 --term HP:0001541 --term HP:0000540 --
↪term HP:0011968 --term HP:0001250 --term HP:0000490 --term HP:0001263 --term␣
↪HP:0001284 --term HP:0002240 --term HP:0001290 --term HP:0031200 --term HP:0011800 --
↪term HP:0000090 --term HP:0000092 --term HP:0001919 --term HP:0012650 --term␣
↪HP:0002419 --term HP:0002119 --term HP:0000105
```

5.5 Multigene inversion

Inversion of ~12.23 kb (`inv(chr3)(9725702; 9737931)`) that disrupts the coding sequence of *BRPF1* was assigned *PSV* score of 8.01.

Pathogenic variants in *BRPF1* are associated with Intellectual developmental disorder with dysmorphic facies and ptosis OMIM:617333.

The phenotypic features curated for this case were:

- HP:0000316 Hypertelorism
- HP:0000494 Downslanted palpebral fissures

- HP:0000431 Wide nasal bridge
- HP:0000286 Epicanthus
- HP:0000311 Round face
- HP:0012368 Flat face
- HP:0000486 Strabismus
- HP:0000508 Ptosis
- HP:0002949 Fused cervical vertebrae
- HP:0002194 Delayed gross motor development
- HP:0000750 Delayed speech and language development
- HP:0002342 Intellectual disability, moderate
- HP:0011150 Myoclonic absence seizure
- HP:0002069 Bilateral tonic-clonic seizure
- HP:0001252 Hypotonia

Data were curated from a published case report Pathogenic 12-kb copy-neutral inversion in syndromic intellectual disability identified by high-fidelity long-read sequencing

5.5.1 Command

```
$ java -jar svanna-cli.jar prioritize -d path/to/svanna-data --vcf example.vcf --term␣
↪HP:0000286 --term HP:0002069 --term HP:0000494 --term HP:0002342 --term HP:0000486 --
↪term HP:0000750 --term HP:0000431 --term HP:0001252 --term HP:0002194 --term␣
↪HP:0012368 --term HP:0011150 --term HP:0002949 --term HP:0000508 --term HP:0000316 --
↪term HP:0000311
```

5.6 Deletion affecting transcription start site

Deletion of 1.57 kb (chrX:64,205,190-64,206,761del) affecting transcription start site of *AMER1* was assigned *PSV* score of 9.05.

Pathogenic variants in *AMER1* are associated with Osteopathia striata with cranial sclerosis (OMIM:300373).

The phenotypic features curated for this case were:

- HP:0001561 Polyhydramnios
- HP:0002684 Thickened calvaria
- HP:0000256 Macrocephaly
- HP:0000316 Hypertelorism
- HP:0031367 Metaphyseal striations
- HP:0002744 Bilateral cleft lip and palate
- HP:0002781 Upper airway obstruction
- HP:0001004 Lymphedema
- HP:0000750 Delayed speech and language development

Data were curated from a published case report Deletion of Exon 1 in AMER1 in Osteopathia Striata with Cranial Sclerosis.

5.6.1 Command

```
$ java -jar svanna-cli.jar prioritize -d path/to/svanna-data --vcf example.vcf --term␣
↪HP:0001561 --term HP:0000750 --term HP:0002684 --term HP:0002781 --term HP:0000316 --
↪term HP:0031367 --term HP:0002744 --term HP:0000256 --term HP:0001004
```

5.7 Deletion affecting promoter region

A deletion of 13 bp (`chr12:6,124,705-6,124,718del`) located in the core promoter region of *VWF* was assigned *PSV* score of 47.26.

In the original publication, the deletion was shown to lead to aberrant binding of Ets transcription factors to the site of the deletion (30 bp upstream of *ENST00000261405.10*) and thereby reduce VWF expression.

Pathogenic variants in *VWF* are associated with von Willebrand disease (`OMIM:193400`).

The phenotypic features curated for this case were:

- `HP:0011890` Prolonged bleeding following procedure
- `HP:0000978` Bruising susceptibility
- `HP:0012147` Reduced quantity of Von Willebrand factor

Data were curated from a published case report Functional characterization of a 13-bp deletion (c.-1522_-1510del13) in the promoter of the von Willebrand factor gene in type 1 von Willebrand disease.

5.7.1 Command

```
$ java -jar svanna-cli.jar prioritize -d path/to/svanna-data --vcf example.vcf --term␣
↪HP:0011890 --term HP:0000978 --term HP:0012147
```

5.8 Translocation disrupting a gene sequence

A translocation (`t(chr3:11,007,014; chr4:139,383,334)`) affecting *SLC6A1* was assigned *PSV* score of 4.51.

Pathogenic variants in *SLC6A1* are associated with Myoclonic-atonic epilepsy (`OMIM:616421`).

The phenotypic features curated for this case were:

- `HP:0000252` Microcephaly
- `HP:0000446` Narrow nasal bridge
- `HP:0000272` Malar flattening
- `HP:0000219` Thin upper lip vermilion
- `HP:0000179` Thick lower lip vermilion
- `HP:0002650` Scoliosis

- HP:0002987 Elbow flexion contracture
- HP:0006380 Knee flexion contracture
- HP:0001250 Seizure
- HP:0001263 Global developmental delay
- HP:0001276 Hypertonia

Data were curated from a published case report Phenotypic consequences of gene disruption by a balanced de novo translocation involving SLC6A1 and NAA15

5.8.1 Command

```
$ java -jar svanna-cli.jar prioritize -d path/to/svanna-data --vcf example.vcf --term␣
↪HP:0000252 --term HP:0000446 --term HP:0000272 --term HP:0000219 --term HP:0000179 --
↪term HP:0002650 --term HP:0002987 --term HP:0006380 --term HP:0001250 --term␣
↪HP:0001263 --term HP:0001263 --term HP:0001276
```