Sniffles2 methods

Moritz Smolka[1], Luis F Paulin[1], Fritz Sedlazeck[1,2,3,4]

[1]Human Genome Sequencing Center, Baylor College of Medicine, Houston, TX, USA;
[2]Department of Molecular and Human Genetics, Baylor College of Medicine, TX, USA;
[3]Aligning Science Across Parkinson’s (ASAP) Collaborative Research Network, Chevy Chase, MD, USA;
[4]Department of Computer Science, Rice University, 6100 Main Street, Houston, TX, USA

Luis F Paulin
Baylor College of Medicine

ABSTRACT

Long-read Structural Variation (SV) calling remains a challenging but highly accurate way to identify complex genomic alterations. Here, we present Sniffles2, which is faster and more accurate than state-of-the-art SV caller across different coverages, sequencing technologies, and SV types. Furthermore, Sniffles2 solves the problem of family- to population-level SV calling to produce fully genotyped VCF files by introducing a gVCF file concept. Across 11 probands, we accurately identified causative SVs around MECP2, including highly complex alleles with three overlapping SVs. Sniffles2 also enables the detection of mosaic SVs in bulk long-read data. As a result, we successfully identified multiple mosaic SVs across a multiple system atrophy patient brain. The identified SV showed a remarkable diversity within the cingulate cortex, impacting both genes involved in neuron function and repetitive elements. In summary, we demonstrate the utility and versatility of Sniffles2 to identify SVs from the mosaic to population levels.

BEFORE START INSTRUCTIONS

Requirements

- Python >= 3.7
- pysam

Tested on:

- Linux CentOS Stream 8
- python==3.9.5
- pysam==0.16.0.1

OCT 02, 2023

OPEN ACCESS

DOI:
dx.doi.org/10.17504/protocols.io.14egn34qyl5d/v1

Protocol Citation: Moritz Smolka, Luis F Paulin, Fritz Sedlazeck 2023. Sniffles2 methods. **protocols.io** https://dx.doi.org/10.17504/protocols.io.14egn34qyl5d/v1

MANUSCRIPT CITATION:
https://www.biorxiv.org/content/10.1101/2022.04.04.487055v2

Protocol status: Working
We use this protocol and it's working

Created: Sep 29, 2023

Last Modified: Oct 02, 2023

PROTOCOL integer ID:
88599

Sniffles2 methodology

1 Installation

Command

Sniffles install pip

```
pip install sniffles
```

Command

Sniffles install conda

```
conda install sniffles=2.2
```

2 Sniffles single sample SV calling

Command

Sniffles call

```
sniffles --input mapped_input.bam --vcf output.vcf.gz --snf output.snf
```

3 Sniffles population SV calling

Command

Sniffles population

```
sniffles --input sample1.bam --vcf sample1.vcf.gz --snf sample1.snf

sniffles --input sample2.bam --vcf sample2.vcf.gz --snf sample2.snf

sniffles --input sampleN.bam --vcf sampleN.vcf.gz --snf sampleN.snf

sniffles --input sample1.snf sample2.snf sampleN.snf --vcf multisample.vcf.gz
```

4 Sniffles low-frequency (mosaic) SV calling

Command

Sniffles mosaic

```
sniffles --input sample.bam --vcf sample_mosaic_sv.vcf.gz --mosaic
```

5 Optional suggested parameters

Command

Sniffles optional suggested

```
# will include the sequence of deletion
--reference reference.fasta

# will output the read names used for every SV
----output-rnames

# will use tandem repeat annotation for the reference genome. Provided for human GRCh37 and GRCh38
--tandem-repeats repeats.bed
```

6 Sniffles genotyping (force-callling) will determine the genotypes for all SVs in the given input .vcf fil

Command

Sniffles genotyping

```
sniffles --input mapped_input.bam --vcf output.vcf.gz --snf output.snf --genotype-vcf known_sv.vcf.gz
```

Long reads alignment

7 **Minimap2**

MINIMAP_PRESET used are: **map-ont** for Oxford Nanopore and **map-hifi** for PacBio HiFi and **map-pb** for PacBio CLR

REFERENCE is either human GRCh37 or GRCh38 with no alt/decoy chromosomes

READS are fastq/compressed-fastq files

OUT is the sample name/identification

Command

Long reads alignment with minimap2 (Linux: CentOS Stream 8)

```
minimap2 \
 -ax ${MINIMAP_PRESET} \
 -t 8  -Y --MD  \
 ${REFERENCE} \
 ${READS} | samtools sort -m 2G - > ${OUT}.bam
```

8 **LRA**

READS are gzip-compressed-fastq files
LRA_PRESET used are: **-ONT** for Oxford Nanopore and **-CCS** for PacBio HiFi and **-CLR**for PacBio CLR
REFERENCE is either human GRCh37 or GRCh38 with no alt/decoy chromosomes
OUTFORMAT is **s** for SAM
OUT is the sample name/identification

Command

Long read alignment with LRA (Linux: CentOS Stream 8)

```
gzip-cd ${READS} | lra align ${LRA_PRESET} -t 8 -p ${OUTFORMAT} --noMismatch ${REFERENCE} /dev/stdin | samtools view -hb - | samtools sort - > ${OUT}.bam"
```

Benchmarking methodology

9

HG002 SV calling for the following technologies and coverage:
Oxford Nanopore Technologies: 5x, 10x, 20x, 30x and 50x
PacBio HiFi: 5x, 10x, 20x, 30x

with **default parameters**

Command

HG002 ONT SV calling default parameters (Linux: CentOS Stream 8)

```
for genome in "grch37" "grch38"; do
   # Sniffles2
   sniffles2 --tandem-repeats human_${genome}.trf.bed -i hg002_ont_${genome}.bam -v sniffles2_hg002_ont_${genome}.vcf -t 8 --reference ${genome}.fasta
   # Sniffles1
   sniffles1 -m hg002_ont_${genome}.bam -v sniffles1_hg002_ont_${genome}.vcf -t 8
   # cuteSV
   cuteSV --max_cluster_bias_INS 100 --diff_ratio_merging_INS 0.3 --max_cluster_bias_DEL 100 --diff_ratio_merging_DEL 0.3 --genotype -t 8 hg002_ont_${genome}.bam ${genome}.fasta cutesv_hg002_ont_${genome}.vcf tmp
   # pbsv
   bash -c "pbsv discover -s hg2 --tandem-repeats human_${genome}.trf.bed hg002_ont_${genome}.bam pbsv.svsig.gz && pbsv call -j 8 ${genome}.fasta pbsv.svsig.gz pbsv_hg002_ont_${genome}.vcf"
   # SVIM
   svim alignment --sequence_alleles tmp hg002_ont_${genome}.bam ${genome}.fasta && mv variants.vcf svim_hg002_ont_${genome}.vcf
done
```

Command

HG002 HiFi SV calling default parameters (Linux: CentOS Stream 8)

```
for genome in "grch37" "grch38"; do
   # Sniffles2
   sniffles2  --tandem-repeats human_${genome}.trf.bed -i hg002_hifi_${genome}.bam -v sniffles2_hg002_hifi_${genome}.vcf -t 8 --reference ${genome}.fasta
   # Sniffles1
   sniffles1  -m hg002_hifi_${genome}.bam -v sniffles1_hg002_hifi_${genome}.vcf -t 8
   # cuteSV
   cuteSV --max_cluster_bias_INS 1000 --diff_ratio_merging_INS 0.9 --max_cluster_bias_DEL 1000 --diff_ratio_merging_DEL 0.5  --genotype -t 8 hg002_hifi_${genome}.bam ${genome}.fasta cutesv_hg002_hifi_${genome}.vcf tmp
   # pbsv
   bash -c "pbsv discover -s hg2 --tandem-repeats human_${genome}.trf.bed hg002_hifi_${genome}.bam pbsv.svsig.gz && pbsv call -j 8 --ccs  ${genome}.fasta pbsv.svsig.gz pbsv_hg002_hifi_${genome}.vcf"
   # SVIM
   svim alignment --sequence_alleles tmp hg002_hifi_${genome}.bam ${genome}.fasta && mv variants.vcf svim_hg002_hifi_${genome}.vcf
done
```

10 **HG002 SV calling** for the following technologies and coverage:
Oxford Nanopore Technologies: 5x, 10x, 20x, 30x and 50x
PacBio HiFi: 5x, 10x, 20x, 30x
with **sensitive parameters**

Command

HG002 ONT SV calling sensitive parameters (Linux: CentOS Stream 8)

```
for genome in "grch37" "grch38"; do
   # Sniffles1
   sniffles1 -s 2 -m hg002_ont_${genome}.bam -v sniffles1_hg002_ont_${genome}.vcf -t 8
   # cuteSV
   cuteSV --max_cluster_bias_INS 100 --diff_ratio_merging_INS 0.3 --max_cluster_bias_DEL 100 --diff_ratio_merging_DEL 0.3 -s 2 --genotype -t 8 hg002_ont_${genome}.bam ${genome}.fasta cutesv_hg002_ont_${genome}.vcf tmp
   # pbsv
   bash -c "pbsv discover -s hg2 --tandem-repeats human_${genome}.trf.bed hg002_ont_${genome}.bam pbsv.svsig.gz && pbsv call -j 8  -A 2 ${genome}.fasta pbsv.svsig.gz pbsv_hg002_ont_${genome}.vcf"
   # SVIM
   svim alignment --sequence_alleles tmp hg002_ont_${genome}.bam ${genome}.fasta && mv variants.vcf svim_hg002_ont_${genome}.vcf
done
```

Command

HG002 HiFi SV calling sensitive parameters (Linux: CentOS Stream 8)

```
for genome in "grch37" "grch38"; do
    # Sniffles1
    sniffles1 -s 2 -m hg002_hifi_${genome}.bam -v sniffles1_hg002_hifi_${genome}.vcf -t 8
    # cuteSV
    cuteSV --max_cluster_bias_INS 1000 --diff_ratio_merging_INS 0.9 --max_cluster_bias_DEL 1000 --diff_ratio_merging_DEL 0.5 -s 2 --genotype -t 8 hg002_hifi_${genome}.bam ${genome}.fasta cutesv_hg002_hifi_${genome}.vcf tmp
    # pbsv
    bash -c "pbsv discover -s hg2 --tandem-repeats human_${genome}.trf.bed hg002_hifi_${genome}.bam pbsv.svsig.gz && pbsv call -j 8 --ccs -A 2 ${genome}.fasta pbsv.svsig.gz pbsv_hg002_hifi_${genome}.vcf"
    # SVIM
    svim alignment --sequence_alleles tmp hg002_hifi_${genome}.bam ${genome}.fasta && mv variants.vcf _hg002_hifi_${genome}.vcf
done
```

11

SV benchmark comparison to Genome in a Bottle SV dataset v0.6 using truvari 2.1 following the GIAB recommended parameters

Command

GRCh37 / GIAB v0.6 SV benchmark (Truvari bench) (Linux: CentOS Stream 8)

```
for longreads in "ont" "hifi"; do
    truvari bench -b HG002_SVs_Tier1_v0.6.vcf.gz -c hg002_${longreads}_grch37.vcf.gz -o truvari_bench_${longreads} -f grch37.fasta --includebed HG002_SVs_Tier1_v0.6.bed --passonly --giabreport
done
```

SV benchmark comparison to Genome in a Bottle SV Challenging Medical Relevant Genes (CMRG) v0.1 using truvari 2.1 following the GIAB recommended parameters

Command

GRCh38 / Challenging Medical Relevant Genes (CMRG) benchmark (Truvari bench) (Linux: CentOS Stream 8)

```
for longreads in "ont" "hifi"; do
   truvari bench -b HG002_CMRG_v0.01.vcf.gz -c hg002_${longreads}_grch38.vcf.gz -o truvari_bench -f grch38.fa --includebed HG002_GRCh38_CRMG_v0.01.bed --passonly
done
```

Simulation of low-frequency SVs

12 **Low-frequency SV simulation**

We used

```
samtools view --subsample
```

to create subset of reads fo HG002 and HG00733 at the following concentrations:

HG002 coverage	HG002 proportion	HG00733 coverage	HG00733 proportion
5	7%	63	93%
7	10%	63	90%
10	14%	60	86%
15	21%	55	79%
20	28%	50	72%

HG002 reads: https://labs.epi2me.io/gm24385_q20_2021.10/
HG002 variants: https://ftp-trace.ncbi.nlm.nih.gov/ReferenceSamples/giab/release/AshkenazimTrio/HG002_NA24385_son/NIST_SV_v0.6/HG002_SVs_Tier1_v0.6.vcf.gz
HG00733 reads: https://www.internationalgenome.org/data-portal/search?q=HG00733
HG00733 variants:
https://ftp.hgsc.bcm.edu/Software/Truvari/3.1/sample_vcfs/hg19/li/HG00733.vcf.gz

13 Read alignment was performed as in **step 7**

14 SV calling with Sniffles2 and cuteSV

Command

Low-frequency SV calling benchmark (Linux: CentOS Stream 8)

```
# Sniffles2 mosaic
sniffles2 --tandem-repeats human_hs37d5.trf.bed -i hg002_hg00733.bam -t 8 --reference grch37.fasta -v sniffles2_hg002_hg00733_mosaic.vcf --mosaic

# Sniffles2 germline (default)
sniffles2 --tandem-repeats human_hs37d5.trf.bed -i hg002_hg00733.bam -t 8 --reference grch37.fasta -v sniffles2_hg002_hg00733_germline.vcf

# cuteSV
cuteSV --max_cluster_bias_INS 100 --diff_ratio_merging_INS 0.3 --max_cluster_bias_DEL 100 --diff_ratio_merging_DEL 0.3 --genotype -t 8 hg002_hg00733.bam  grch37.fasta  cute_SV_hg002_hg00733.vcf tmp
```

15 SV benchmark
We then used the SV from the GIAB v0.6 benchmark (see **step 11**) and compared the three call sets: Sniffles germline, Sniffles mosaic and cuteSV. For Sniffles mosaic we also filtered GIAB v0.6 benchmark based on the variant allele frequency (VAF) range that Sniffles2 mosaic mode uses (VAF 5%-20%) to compute the adjusted recall.

Mendelian inconsistency benchmark in population mode

16 HG002 reads: https://labs.epi2me.io/gm24385_q20_2021.10/
HG003 reads:
https://ftp.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG003_NA24149_father/UCSC_Ultralong_OxfordNanopore_Promethion/
HG003 reads:

https://ftp.ncbi.nlm.nih.gov/ReferenceSamples/giab/data/AshkenazimTrio/HG004_NA24143_mother/UCSC_Ultralong_OxfordNanopore_Promethion/

17 Read alignment was don as in **step 7**

18 Sniffles2 SV calling was done as in **step 3**

19 cuteSV SV calling was done as in **step 10** for Oxford Nanopore data for each sample
Next SURVIVOR was used to merge the SV calls from the three samples

Command

SURVIVOR merge (Linux: CentOS Stream 8)

```
ls cutesv_hg002_grch37.vcf cutesv_hg003_grch37.vcf cutesv_hg004_grch37.vcf > trio_samples.list
survivor merge trio_samples.list 1000 1 1 0 0 50 cuteSV_trio_grch37.vcf
```

20 An additional step was performed for cuteSV which consists in genotyping/force-calling the SV from the merged VCF, to then merge again with SURVIVOR

Command

cuteSV force-call and merge (Linux: CentOS Stream 8)

```
# Calling
for sample_id in "hg002" "hg003" "hg004"; do
   cuteSV --max_cluster_bias_INS 100 --diff_ratio_merging_INS 0.3 --max_cluster_bias_DEL 100 --diff_ratio_merging_DEL 0.3 -s 2 --genotype -t 8 ${sample_id}_grch37.bam grch37.fasta cutesv_${sample_id}_grch37_force.vcf tmp -Ivcf cuteSV_trio_grch37.vcf
done

# Merge
ls cutesv_hg002_grch37_force.vcf cutesv_hg003_grch37_force.vcf cutesv_hg004_grch37_force.vcf > trio_samples_force.list
survivor merge trio_samples_force.list 1000 1 1 0 0 50 cuteSV_trio_grch37_force.vcf
```

21 Mendelian consistency test with **bcftools's mendelian plugin** with the three output files: Sniffles2 population merge (**step 18**), cuteSV (**step 19**) and cuteSV force-called (**step 20**)

Command

bcftools mendelian consistency (Linux: CentOS Stream 8)

```
# Sniffles2 population
bcftools +mendelian sniffles2_trio_grch37.vcf.gz -t hg004,hg003,hg002

# cuteSV vanilla
bcftools +mendelian cuteSV_trio_grch37.vcf -t hg004,hg003,hg002

# cuteSV force-called
bcftools +mendelian cuteSV_trio_grch37_force.vcf -t hg004,hg003,hg002
```

Chromosome X disorder patient analysis

22 We called all samples as in **step 3** (Sniffles2 population SV calling) and subsequently merged them (**step 3**). We used the resulting fully-genotyped population VCF file of the rest of the analysis.

Next, we filtered out SV if they were in either of the following categories:

- SV < 10kb
- SV present in either a mother or father based on the sample identification and SUPP_VEC (see below)
- SV in an autosome

We ended up with SV that were only present in the probands and that were likely causative of the observed phenotype.
Filtering was done with bcftools view and a custom python script "**sniffles2_vcf_parser.py**"found in https://github.com/smolkmo/Sniffles2-Supplement and https://zenodo.org/record/8122060

The **SUPP_VEC** is a field in the INFO section of the VCF file that denotes the presence/absence of a genetic variant (in our case Structural Variant, SV) in the sample for the sample position/index
For samples A B and C, the **SUPP_VEC=101** means that the genetic variant is present in samples A and C