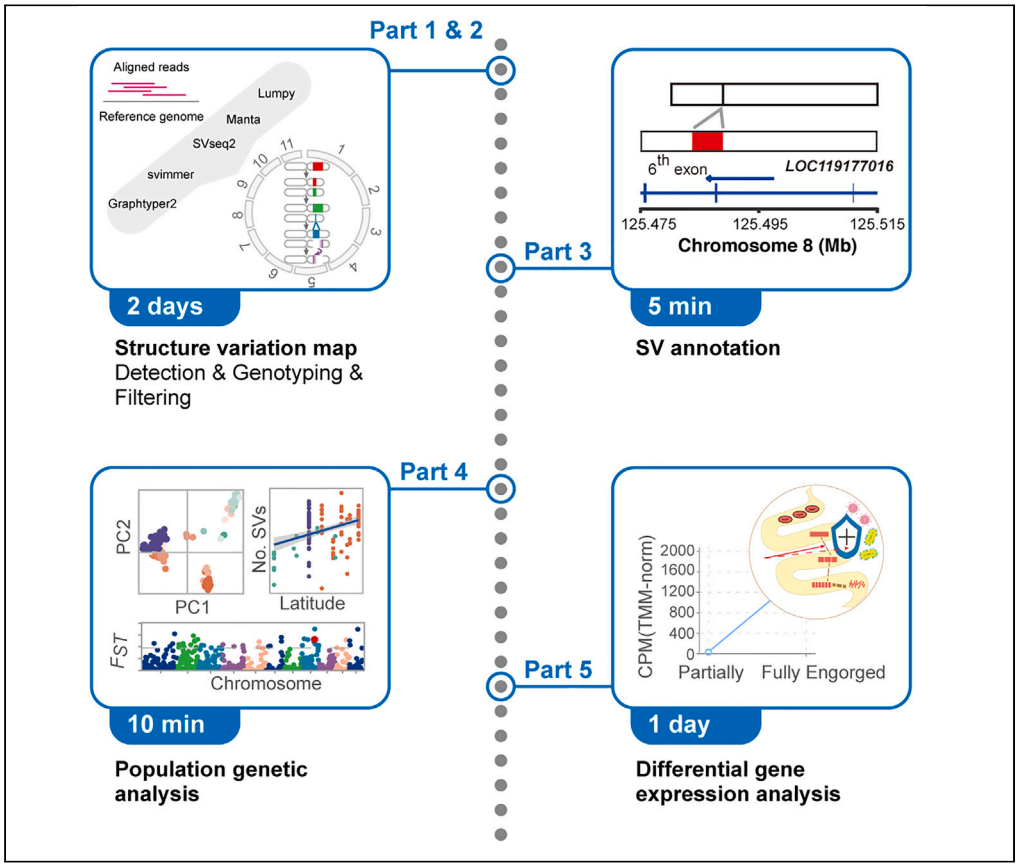
STAR Protocols

CellPress
OPEN ACCESS

Protocol

A protocol for applying low-coverage whole-genome sequencing data in structural variation studies



Structural variations (SVs) have a great impact on various biological processes and influence physical traits in many species. Here, we present a protocol for applying the low-coverage next-generation sequencing data of *Rhipicephalus microplus* to detect high-differentiated SVs accurately. We also outline its use to investigate population/species-specific genetic structures, local adaptation, and transcriptional function. We describe steps for constructing variation maps and SV annotation. We then detail population genetic analysis and differential gene expression analysis.

Publisher's note: Undertaking any experimental protocol requires adherence to local institutional guidelines for laboratory safety and ethics.

Qi Liu, Bo Xie, Yang Gao, Shuhua Xu, Yan Lu

xushua@fudan.edu.cn (S.X.)
lueyan@fudan.edu.cn (Y.L.)

Highlights

A protocol for SV detection and genotyping using low-coverage NGS data

Analytic procedures for characterizing species-specific genetic architectures

A workflow for discovering potential SVs associated with important phenotypes

Liu et al., STAR Protocols 4, 102433
September 15, 2023
https://doi.org/10.1016/j.xpro.2023.102433

STAR Protocols

CellPress
OPEN ACCESS

Protocol

A protocol for applying low-coverage whole-genome sequencing data in structural variation studies

Qi Liu,[1,4] Bo Xie,[2] Yang Gao,[1,2,3] Shuhua Xu,[1,2,3,*] and Yan Lu[1,5,*]

[1]State Key Laboratory of Genetic Engineering, Center for Evolutionary Biology, Human Phenome Institute, Zhangjiang Fudan International Innovation Center, Collaborative Innovation Center of Genetics and Development, School of Life Sciences, Fudan University, Shanghai 201203, China

[2]Key Laboratory of Computational Biology, Shanghai Institute of Nutrition and Health, University of Chinese Academy of Sciences, Chinese Academy of Sciences, Shanghai 200031, China

[3]School of Life Science and Technology, Shanghai Tech University, Shanghai 201210, China

[4]Technical contact: liuqi_@fudan.edu.cn

[5]Lead contact

*Correspondence: xushua@fudan.edu.cn (S.X.), lueyan@fudan.edu.cn (Y.L.)
https://doi.org/10.1016/j.xpro.2023.102433

SUMMARY

Structural variations (SVs) have a great impact on various biological processes and influence physical traits in many species. Here, we present a protocol for applying the low-coverage next-generation sequencing data of *Rhipicephalus microplus* to detect high-differentiated SVs accurately. We also outline its use to investigate population/species-specific genetic structures, local adaptation, and transcriptional function. We describe steps for constructing variation maps and SV annotation. We then detail population genetic analysis and differential gene expression analysis.

For complete details on the usage and execution of this protocol, please refer to Liu et al. (2023).

BEFORE YOU BEGIN

Download the test dataset

Timing: 1 day

1. For a rapid test of the first and second parts of this protocol, we use the publicly available pair-end next-generation sequencing (NGS) data of three *Rhipicephalus microplus* (*R. microplus*) as the example data[1] and calculate the time of execution for each step.

   Note: We provide three bam files for testing, including chromosome 11 of three tested samples, which are freely available at GitHub: https://github.com/Shuhua-Group/SVanalysis_STARProtocols.

2. In this protocol, the reference genome BIME_Rmic_1.3 (*R. microplus*) is used and downloaded from NGDC (NGDC: PRJCA002240) or NCBI (BioProject: PRJNA633311) database. Here, we download the reference genome FASTA file (GWHAMMN00000000.genome.fasta.gz) from NGDC, and the chromosome identifier in the FASTA file is a character (e.g., GWHAMMN00000001).

   Note: If the chromosome identifier in the FASTA file (GCF_013339725.1_ASM1333972v1_genomic.fna.gz) from NCBI is a number (e.g., 1), you can convert it by following the relationship listed in Table 1 or using our script "code_convertChrIdentifier.pl". Make sure that the chromosome identifier is consistent throughout all analyses, e.g., "GWHAMMN00000001" or "1".

Table 1. The information about the reference genome of *R. microplus*

Chromosome label (NGDC)	Chromosome number (NCBI)	Chromosome length
GWHAMMN00000001	1	325144201
GWHAMMN00000004	2	202141382
GWHAMMN00000005	3	238537928
GWHAMMN00000006	4	218457896
GWHAMMN00000007	5	206569738
GWHAMMN00000008	6	183350851
GWHAMMN00000009	7	175432524
GWHAMMN00000010	8	170387423
GWHAMMN00000011	9	155099419
GWHAMMN00000002	10	139434689
GWHAMMN00000003	11	126224405

3. In addition, we provide GFF and GTF files generated by our group, which contain gene structure and details supplied by the TIGMIC group and the NCBI RefSeq database (See key resources table).

Download the software and scripts

Timing: 1–2 days

4. The majority of the analyses in this protocol are performed with the aid of already-available software, which is listed in the key resources table and can be downloaded via the links provided.
5. We also create a Docker image (svanalysis_starprotocols:1.0.0) that provides an environment with all software installed.

Note: The usage of the docker image is available at GitHub: https://github.com/Shuhua-Group/SVanalysis_STARProtocols.

6. We also offer analytic script files for variant filtering, combining different genome annotations, and variant annotation which are available on GitHub (See key resources table).
7. We offer scripts to sequentially run each of the aforementioned programs, resulting in three pipelines (See key resources table).

Compile a list of phenotype-related genes (optional)

Timing: 1 min

8. Jia et al.[1] provides a list of genes associated with tick hematophagy and the related phenotype (Table S3 in Jia et al., 2020[1]), which includes 1,028 genes in *R. microplus*. You can also curate a list of phenotype-associated genes related to your trait of interest. The number of genes on the list has no set limit. This step is optional.

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
AHRm.725 in Central China	Jia et al.[1]	https://ngdc.cncb.ac.cn/gsa/browse/CRA002715; ftp://download.big.ac.cn/gsa3/CRA002715/CRR142483/CRR142483_f1.fq.gz; ftp://download.big.ac.cn/gsa3/CRA002715/CRR142483/CRR142483_r2.fq.gz

(*Continued on next page*)

Continued

REAGENT or RESOURCE	SOURCE	IDENTIFIER
YNRm.729 in Southwest China	Jia et al.[1]	https://ngdc.cncb.ac.cn/gsa/browse/CRA002715; ftp://download.big.ac.cn/gsa3/CRA002715/CRR142576/CRR142576_f1.fq.gz; ftp://download.big.ac.cn/gsa3/CRA002715/CRR142576/CRR142576_r2.fq.gz
HnRm.648 in South China	Jia et al.[1]	https://ngdc.cncb.ac.cn/gsa/browse/CRA002715; ftp://download.big.ac.cn/gsa3/CRA002715/CRR142531/CRR142531_f1.fq.gz; ftp://download.big.ac.cn/gsa3/CRA002715/CRR142531/CRR142531_r2.fq.gz
Rhipicephalus microplus genome (GWHAMMN00000000.genome.fasta.gz)	Jia et al.[1]	https://bigd.big.ac.cn/gwh/Assembly/8870/show; https://ftp.ncbi.nlm.nih.gov/genomes/all/GCF/013/339/725/GCF_013339725.1_ASM1333972v1/GCF_013339725.1_ASM1333972v1_genomic.fna.gz
7 *R microplus* transcriptomes	Tirloni et al.[2]	https://www.ncbi.nlm.nih.gov/bioproject/PRJNA232001
Rhipicephalus_microplus.NCBI.TIGMIC.sorted.updPos.gff.gz	In this protocol	https://github.com/Shuhua-Group/SVanalysis_STARProtocols; https://zenodo.org/record/7794021#.ZCqB_exByIE; https://pog.fudan.edu.cn/#/software;
Rhipicephalus_microplus.NCBI.TIGMIC.sorted.updPos.gtf.gz	In this protocol	https://github.com/Shuhua-Group/SVanalysis_STARProtocols; https://zenodo.org/record/7794021#.ZCqB_exByIE; https://pog.fudan.edu.cn/#/software
Rhipicephalus_microplus.SR.LC.SD.mask.0based.bed	In this protocol	https://github.com/Shuhua-Group/SVanalysis_STARProtocols; https://zenodo.org/record/7794021#.ZCqB_exByIE; https://pog.fudan.edu.cn/#/software
Software and algorithms		
BWA (v0.7.17)	Li et al.[3]	https://bio-bwa.sourceforge.net/
GATK (v4)	McKenna et al.[4]	https://gatk.broadinstitute.org/hc/en-us
Picard (v2.26.11)	McKenna et al.[4]	https://broadinstitute.github.io/picard/
Manta (v1.6.0)	Chen et al.[5]	https://github.com/Illumina/manta
Lumpy (v0.2.13)	Layer et al.[6]	https://github.com/arq5x/lumpy-sv
SVseq2 (v2.2)	Zhang et al.[7]	https://sourceforge.net/projects/svseq2/files/SVseq2_2/
svimmer (v0.1)	Eggertsson et al.[8]	https://github.com/DecodeGenetics/svimmer
Graphtyper2 (v2.7.4)	Eggertsson et al.[8]	https://github.com/DecodeGenetics/graphtyper
Vcflib	Garrison et al.[9]	https://github.com/vcflib/vcflib
RepeatMasker (open-4.0.9)	Tarailo-Graovac and Chen[10]	https://www.repeatmasker.org/
RepeatModeler (v2.0.1)	Tarailo-Graovac and Chen[10]	https://www.repeatmasker.org/
sedef (v1.1)	Numanagic et al.[11]	https://github.com/vpc-ccg/sedef
BEDTools (v 2.30.0)	Quinlan et al.[12]	https://bedtools.readthedocs.io/en/latest/
SAMtools (v1.17)	Danecek et al.[13]	https://www.htslib.org/download/
BCFtools (v1.17)	Danecek et al.[13]	https://www.htslib.org/download/
VCFtools (v0.1.16)	Danecek et al.[14]	https://github.com/vcftools/vcftools
PLINK (v1.9, v2.0)	Chang et al.[15]	https://www.cog-genomics.org/plink/
FastQC (v0.12.1)		https://github.com/s-andrews/FastQC
Trim_galore (v0.6.7)		https://github.com/FelixKrueger/TrimGalore
HISAT2 (v2.2.1)	Kim et al.[16]	https://github.com/DaehwanKimLab/hisat2
HTSeq (v2.0.2)	Anders et al.[17]	https://github.com/htseq/htseq
dplyr package (v1.1.1)	Wickham et al.[18]	https://CRAN.R-project.org/package=dplyr
EdgeR package (v3.38.2)	Robinson et al.[19]	https://bioconductor.org/packages/release/bioc/html/edgeR.html
GFOLD (v1.1.4)	Feng et al.[20]	https://github.com/knowledgefold/gfold
Scripts for data analysis	In this protocol	https://github.com/Shuhua-Group/SVanalysis_STARProtocols; https://pog.fudan.edu.cn/#/software
Other		
Linux server	N/A	N/A

MATERIALS AND EQUIPMENT

- Reference genome assemblies (FASTA file) and the short-read sequences (FASTQ file) of samples (See download the test dataset section before you begin).

- Software and scripts used in this protocol (See the software and algorithms section of the key resources table).
- A basic knowledge of scripting languages (R, Python, Perl, and bash), as well as software (BEDTools, PLINK, VCFtools, and BCFtools) is required to understand and apply this protocol.
- All tests are run on 64-core Intel Xeon CPU E3-1280 v5 3.70 GHz Linux servers. We recommend to using computing clusters to perform this protocol. For SV detection and genotyping, assuming 12 CPUs, at least 30 GB of RAM is required.

STEP-BY-STEP METHOD DETAILS

Part 1: SV detection from the short-read sequences with low coverage

Timing: 1 day (12 threads, variable depending on the number of threads used)

The discovery of reliable structure variation (SV), including deletion (DEL), duplication (DUP), insertion (INS), and inversion (INV), in each sample is the first and most important step in SV studies. There are four basic short-read sequencing-based methods: read pairs (RP), read depth (RD), split read (SR), assembly (AS), as well as a combination of the aforementioned ones (CB). Kosugi et al. (2019) demonstrated that the combination of specific algorithms (SV detection tools using one or more aforementioned basic methods) effectively improves SV detection accuracy.[21] Then, we first employ three algorithms, including Manta (RP-SR-AS), Lumpy (RP-SR-RD), and SVseq2 (SR), to discover reliable SVs from the short-read sequences with low coverage (4–10×). The testing is only depicted in the following code shows on one individual (AHRm.725), but the same procedure can be applied to all samples as well.

1. Prepare the config file to serve as the input file of our protocol.

```
# The config file contains five parts

# Please don't change the variable name in the config file. The variable name will be used in our pipelines.

# The following path of software are paths in our Docker image.

# Owing to failed to install pacakge 'pysam', we do not install HTSeq and lumpy-sv in our docker image

#======== Please Copy the following information into a config file ========

# Part 1: The path of executive program to be used

#**** common executive program

bgzip=/root/software/samtools-1.17/htslib-1.17/bgzip

tabix=/root/software/samtools-1.17/htslib-1.17/tabix

bcftools=/usr/local/bin/bcftools

vcftools=/usr/local/bin/vcftools

samtools=/usr/local/bin/samtools

perl=/usr/bin/perl

java=/usr/bin/java

picard=/root/software/picard_2.26.11/picard.jar

python3_9=/usr/bin/python3

#**** add path of bgzip into PATH
```

```
export PATH=${PATH}:/root/software/samtools-1.17/htslib-1.17

# Python2 be used to run Manta and Lumpy

python2=/usr/bin/python2

# Part 2: The path of executive program to be used in SV calling

#**** 110.raw.reads.mapping

bwa=/usr/bin/bwa

gatk=/root/software/gatk-4.4.0.0/gatk

#**** 130.sv.detection

manta_install_path=/root/software/manta-1.6.0.centos6_x86_64

# User can install lumpy-sv using conda with command: /path miniconda3/bin/conda install
lumpy-sv=0.2.13

lumpy_install_path=/your_software_path/lumpy-sv-0.2.13

#! The format of inversion detected by Manta should be converted to another format using the
following script

#https://github.com/Illumina/manta/tree/master/src/python/libexec/convertInversion.py

code_convertInversion=/root/software/manta-1.6.0.centos6_x86_64/libexec/
convertInversion.py

svseq2=/root/software/SVseq2/SVseq2_2

svimmer=/root/software/svimmer-master/svimmer

#**** 140.sv.genotyping

graphtyper2=/root/software/graphtyper_2.7.4/graphtyper

#**** 150.sv.filtering

vcffilter=/usr/bin/vcffilter

#**** downstream analysis

bedtools=/root/software/bedtools_v2.30.0/bedtools

plink19=/root/software/plink_1.9/plink

plink2=/root/software/plink_2/plink2

# Part 3: The path of executive program to be used in gene expression

#**** 110.hisat2.index_genome

hisat2_build=/root/software/hisat2-master/hisat2-build

#**** 120.hisat2.genome_out

hisat2=/root/software/hisat2-master/hisat2

#**** 130.htseq.genome_out

# User can install htseq using conda with command: /path/miniconda3/bin/conda install
htseq=2.0.2

htseq_count=/your_software_path/python-3.7.13/bin/htseq-count

# R version 3.6.3

Rscript=/usr/bin/Rscript

# We have install the following packages in Docker: dplyr, edgeR

# Make sure you have installed these packages in yourself system before testing this protocol.
```

```
# Part 4: The path of input data to be used in SV analysis

#****/ The directory of scripts in our protocol

this_protocol_script_path=/your_software_path/SVprotocol

#****/ The directory of input and output data to be used

your_analysis_dir=/your_data_path/SVCalling

#** In your_analysis_dir (/your_data_path/SVCalling), you must have a subdirectory named "fastq/<sample_name>" including two FASTQ files (<sample_name>_R1.fastq.gz and <sample_name>_R2.fastq.gz)

# For example, for sample AHRm.725, we required two files in the following directory

# $your_analysis_dir/fastq/AHRm.725/AHRm.725_R1.fastq.gz

# $your_analysis_dir/fastq/AHRm.725/AHRm.725_R2.fastq.gz

#** When you source this config, this config will create some subdirectory in your_analysis_dir, which will be the directory of some output data.

#****/

mkdir -p $your_analysis_dir/110.raw.reads.mapping 2>/dev/null

mkdir -p $your_analysis_dir/120.remove.duplicates 2>/dev/null

mkdir -p $your_analysis_dir/130.sv.detection 2>/dev/null

mkdir -p $your_analysis_dir/140.sv.genotyping 2>/dev/null

mkdir -p $your_analysis_dir/150.sv.filtering 2>/dev/null

#****/ The full path of reference genome sequence data (FASTA)

#** Please just include chromosome-level sequence

#** Reference genome data must end with "fasta"

#** Please put all reference data into one directory

#****/

ref_genome_fasta=/your_bundle_path/reference/Rhipicephalus_microplus.chromosome.fasta

# Part 5: The path of input data to be used in gene expression analysis

#****/ The directory of input and output data to be used

#** In your_gene_expression_analysis_dir, you must have a subdirectory named "fastq/<sample_name>" including two FASTQ files (<sample_name>_1.fastq.gz and <sample_name>_2.fastq.gz) for pair-end sequencing data

#****/

your_gene_expression_analysis_dir=/ your_gene_expression_path/gene_express

# Files used in script ''SVprotocol_part5.sh''

ref_genome_gtf=/your_bundle_path/reference/
Rhipicephalus_microplus.NCBI.TIGMIC.sorted.updPos.gtf

ref_genome_gff=/your_bundle_path/reference/
Rhipicephalus_microplus.NCBI.TIGMIC.sorted.updPos.gff.gz

# A variable used in script ''SVprotocol_part5.sh'' as the prefix of index files of the reference genome, which index file is used in hisat2

ref_genome_prefix_name=Rhipicephalus_microplus
```

2. Prepare pair-end sequencing data as the input file of our protocol "SVprotocol_part1.sh".

```
#****/ The directory of input and output data to be used

your_analysis_dir=/your_data_path/SVCalling

# In the config file, we have added the above information. So, in your_analysis_dir, the user must make a subdirectory named "fastq/<sample_name>" including two FASTQ files (<sample_name>_R1.fastq.gz and <sample_name>_R2.fastq.gz). For example, for sample AHRm.725, we required two files in the following directory

$your_analysis_dir/fastq/AHRm.725/AHRm.725_R1.fastq.gz

$your_analysis_dir/fastq/AHRm.725/AHRm.725_R2.fastq.gz
```

3. Index the reference genome file.

```
#****/ The full path of reference genome sequence data (FASTA)

#** Please just include chromosome-level sequence

#** Reference genome data must end with ``fasta''

#****/

ref_genome_fasta=/your_bundle_path/reference/Rhipicephalus_microplus.chromosome.fasta

# In the config file, we have added the above information, So, the user can index the reference genome fasta as follows:

>cd /your_bundle/path/reference

>bwa index Rhipicephalus_microplus.chromosome.fasta
```

4. Map the pair-end short-read sequence to the reference genome and remove duplicated reads.

Note: Make sure that two FASTQ files for the same sample (*_R1.fastq.gz and *_R2.fastq.gz) are in the same directory named "fastq/*sample_name". It is recommended that the genome sequence file (FASTA) only contain the sequence at the chromosome level, i.e., chromosomes 1–11.

5. Detect SVs in each sample using three algorithms (Manta, Lumpy, and SVseq2).
   a. Detect SVs using Manta (troubleshooting 1).
   b. Detect SVs using Lumpy (troubleshooting 1).
   c. Detect SVs using SVseq2.
      i. Divide an individual's bam file by chromosome and separately detect deletions in each chromosome using SVseq2 (troubleshooting 2 and 3)

⚠ CRITICAL: SVseq2 can only use an uncompressed reference genome fasta file with the suffix "fasta" as input (troubleshooting 4).

      ii. Merge the results throughout all chromosomes and save the outputs as a VCF file.
6. Use the svimmer software to merge the SVs discovered by Manta, Lumpy, and SVseq2 in each sample and retain the SVs with a size range between [50 bp, 2 Mb].

Note: Manta can discover DEL, INS, INV, and DUP. Lumpy is capable of finding DEL, INV, and DUP. SVseq2 can discover DEL. Only variants that were found by at least two tools were retained, except for insertion, which was found by Manta. The users can modify the filtering criteria in our pipeline (code_SV_filtering.pl) with the parameters "–len" and "–xlen".

```
# We have generated a pipeline including all necessary processes (steps 4-6) to detect SVs for one sample

# Please read the usage before you run this pipeline

# Use the above config file following the argument ''--config''

>sh /this_protocol_script_path/SVprotocol_part1.sh --sample AHRm.725 -config /this_protocol_script_path/example.config --thread 12
```

Part 2: SV genotyping and filtering

Timing: 6 h (12 threads, variable depending on the number of threads and samples used)

Population-scale genotyping is conducted by Graphtyper2. Next, we filter out SVs that match the criteria suggested by Graphtyper2. Additionally, we eliminate SVs that overlap with low-complexity regions, simple repeats, DNA satellites, or regions of segmental duplications, as well as those with large genotyping missing rates.

7. For all samples, we first merge SVs detected through the above steps using the svimmer software while keeping SVs with a size range between [50 bp, 2 Mb].

   Note: The users can modify the filtering criteria in our pipeline (code_SV_filtering.pl) with the parameters "–len" and "–xlen".

8. To improve the accuracy of SV genotyping, we first use SAMtools to generate a file containing the average coverage divided by the read length for each bam/sample (one value per line), and then subsample reads in such regions using argument "avg_cov_by_readlen" in Graphtyper2.

   Note: When reads are mapped to the reference genome, some genomic regions have abnormally high read coverage (e.g., 10 times the average coverage), which will have an impact on the accuracy of genotyping.

9. Genotype SVs for each chromosome using Graphtyper2.
10. Concatenate SV genotyping results throughout all chromosomes and save outputs as a VCF file.
11. Filter SVs.
    a. Filter out SVs followed by the recommendation suggested by Graphtyper2 with the command "(SVTYPE = DEL & QD > 12 & (ABHet > 0.30 | ABHet < 0) & (AC / NUM_MERGED_SVS) < 25) | (SVTYPE = DUP & QD > 5 & (AC / NUM_MERGED_SVS) < 25) | (SVTYPE = INS & (AC / NUM_MERGED_SVS) < 25 & (ABHet > 0.25 | ABHet < 0) & MaxAAS > 4) | (SVTYPE = INV & (AC / NUM_MERGED_SVS) < 25 & (ABHet > 0.25 | ABHet < 0) & MaxAAS > 4)".

    Note: Graphtyper2 classifies SV genotype into four categories, i.e., "PASS" with genotype quality (GQ) $\geq$ 30, "FAIL1" with GQ $\geq$ 20, "FAIL2" with GQ $\geq$ 10, and "FAIL3" with GQ < 10. "PASS_AC" means the total number of alternate alleles in the called genotype, and "PASS_ratio" means the ratio of genotype with the "PASS" flag. Considering the sequencing data with low coverage, we remove the filtering argument "PASS_AC" and "PASS_ratio" in our protocol.

    b. Set SV genotype as missing genotype if the read depth (DP) < 1 or genotype quality (GQ) < 13. GQ is larger or equal to 13 means that the error rate of genotype calling is less than 5% (95% confidence). Considering the low coverage of the sequencing data, we relax the DP and GQ criteria in our protocol.

Note: The users can modify the filtering criteria in our pipeline (code_convert_GT_FT.pl) with the parameters "–dp" and "–gq". For example, GQ ≥ 10 (90% confidence), GQ ≥ 15 (97% confidence), GQ ≥ 20 (99% confidence), and GQ ≥ 30 (99.9% confidence).

c. Filter out SVs with genotyping missing rates larger than 50%.

Note: Considering the low coverage of the sequencing data, we relax the genotyping missing rate criteria in our protocol. Users can use other criteria in the initial filtration based on the data quality, e.g., 40%.[22] If the NGS data have read coverage larger than 10, users can skip steps "b" and "c" and modify the command in our pipeline to the following command: "(SVTYPE = DEL & QD > 12 & (ABHet > 0.30 | ABHet < 0) & (AC / NUM_MERGED_SVS) < 25 & PASS_AC > 0 & PASS_ratio > 0.1) | (SVTYPE = DUP & QD > 5 & PASS_AC > 0 & (AC / NUM_MERGED_SVS) < 25) | (SVTYPE = INS & PASS_AC > 0 & (AC / NUM_MERGED_SVS) < 25 & PASS_ratio > 0.1 & (ABHet > 0.25 | ABHet < 0) & MaxAAS > 4) | (SVTYPE = INV & PASS_AC > 0 & (AC / NUM_MERGED_SVS) < 25 & PASS_ratio > 0.1 & (ABHet > 0.25 | ABHet < 0) & MaxAAS > 4)".

d. Filter out SVs that overlap with low-complexity regions, simple repeats, DNA satellites, or regions of segmental duplications. Firstly, we calculate the cumulative size of overlapped regions between a SV with 1 or N repeat regions (1-bp overlap threshold). Then the ratio of the cumulative size of overlapped regions is calculated. Finally, SVs with a ratio larger or equal to 0.5 are filtered out.[22]
   i. Create a *de novo* repeat database using RepeatModeler software. The following code only shows one chromosome, but the same process can be used for other chromosomes as well.

Note: Make sure that the sequence file for each chromosome has been prepared individually before analysis (troubleshooting 5).

```
# For one chromosome
# BuildDatabase -name <species>_<chr> <species>.<chr>.unmasked.fasta
# -database <species>_<chr>
# You can change the threads through the argument ``-pa''
# Chromosome: GWHAMMN00000001 == 1
>BuildDatabase -name Rhipicephalus_microplus_1 Rhipicephalus_microplus.1.unmasked.fasta
>RepeatModeler -pa 10 -database Rhipicephalus_microplus_1 -LTRStruct
# After completing analysis for all chromosomes
>cat Rhipicephalus_microplus_*-families.fa > Rhipicephalus_microplus.genome.denovo.repeat.lib
```

   ii. Identify repeat sequences using the RepeatMasker. Only one chromosome is displayed in the code, but the procedure is the same step for all chromosomes.

```
# For one chromosome
# <species>.<chr>.unmasked.fasta
# <species>.genome.denovo.repeat.lib
>RepeatMasker -xsmall -lib Rhipicephalus_microplus.genome.denovo.repeat.lib -pa 10 Rhipicephalus_microplus.1.unmasked.fasta
# After completing analysis for all chromosomes
# Merge all masked fasta to one fasta
```

```
>cat *fasta.masked > Rhipicephalus_microplus.gwd.repeatmasker.
masked.fasta

# Extract the position and sequence type for each repeat region

>for j in `ls *unmasked.fasta.out`

do

sed '1,3d' $j | awk 'BEGIN{OFS="\t"}{print $5,$6,$7,$10,$11}' >> Rhipicephalus_microplus.
gwd.repeatmasker.bed

done

# Extract simple repeat or low complexity region for the downstream analysis

>awk '{if($5=="Simple_repeat" || $5=="Low_complexity")print $0}' Rhipicephalus_micro-
plus.gwd.
repeatmasker.bed > Rhipicephalus_microplus.gwd.repeatmasker.simpleRepeat_lowComplex.bed
```

iii. Identify segment duplication regions using sedef software.

```
# You can change the threads through the argument ''-j''

>sedef.sh -o Rhipicephalus_microplus -j 20 Rhipicephalus_microplus.chromosome.fasta
```

iv. Combine the bed files generated in the above steps, then use this file as an input file for our pipeline with the argument "–repeat".

```
# Prepare a file named ''sample.list'' in the following format:

AHRm.725

HnRm.648

YNRm.729

# We have generated a pipeline including all necessary processes (steps 7–11) to genotype and
filter SVs for all samples

# Please read the usage before you run this pipeline

# Use the above config file following the argument ''--config''

>sh /this_protocol_script_path/SVprotocol_part2.sh --samplelist sample.list --config /this_
protocol_script_path/example.config --thread 12 --repeat Rhipicephalus_microplus.SR.LC.SD.
mask.0based.bed
```

Part 3: SV annotation

Timing: 5 min

In this part, we will annotate SVs to gene regions using GFF3 or GTF files.

12. Convert GFF3 or GTF to a 0-based bed-format file (Figure 1, columns 11–15) including five columns: <chromosome><start><end><geneName_functionalRegion><functionalRegion>. The functional region includes the coding region (CDS), promoter region, intronic region, intergenic region, and so on.

Column1 SV_chr	Column2 SV_start	Column3 SV_end	Column4 SV_name	Column5 SV_type	Column6 SV_size
GWHAMMN00000001	409717	526955	GWHAMMN00000001:409718:UG	DUP	117237
GWHAMMN00000001	1287624	1821061	GWHAMMN00000001:1287625:UG	DUP	533436
GWHAMMN00000001	1287624	1821061	GWHAMMN00000001:1287625:UG	DUP	533436

Column7 num_of_ind_haveSV	Column8 SV_count_in_pop	Column9 num_of_ind_haveCall	Column10 num_of_ind_in_POP	Column11 gene_feature_chr
1	1	113	138	GWHAMMN00000001
2	4	110	138	GWHAMMN00000001
2	4	110	138	GWHAMMN00000001

Column12 gene_feature_start	Column13 gene_feature_end	Column14 gene_name:gene_feature	Column15 gene_feature
512058	512266	LOC119167497:CDS	CDS
1377354	1378812	HPB51_020232:CDS	CDS
1376354	1377354	HPB51_020232:promoter	Promoter

Figure 1. Screenshot of the output files in SV annotation analysis using BEDTools software
The information above the dashed line represents the meaning of each column.

Note: The user can extend the definition of functional regions, e.g., upstream and downstream regions.

13. Screen functional regions existed in the above bed-format file for overlap with SVs using the BEDTools software. A file generated by our protocol with the suffix "perSite.SVcount" can be used as the input file of BEDTools with the argument "-a" (Figure 1).

```
# Convert GFF to 0-based bed file

>perl  /this_protocol_script_path/code_gff2SVAnnBed.pl  Rhipicephalus_microplus.NCBI.TIGMIC.sorted.updPos.gff.gz out_prefix

# In the subdirectory ``140.sv.genotyping'', you can find a 1-based bed file named ``population.graphtyper2.aggregated.PASS.GQ13.DP1.GENO0.5.rmRepeat.perSite.SVcount''

# Tab-separated Format: <chr><start><end><ID_in_VCF><SV_type><SV_size><num_of_individual_haveSV><SV_count_in_population><num_of_individual_haveGenotypeCall><num_of_individual_in_population>

# Before running BEDTools, convert 1-based bed file to 0-based bed file

>input_sv_bed= population.graphtyper2.aggregated.PASS.GQ13.DP1.GENO0.5.rmRepeat.perSite.SVcount

>awk 'BEGIN{OFS="\t"}{$2=$2-1;print $0}' $input_sv_bed > ${input_sv_bed}.bed

>bedtools intersect -a ${input_sv_bed}.bed -b out_prefix.gff.cds.bed -wa -wb > sv.annotation.bed
```

Part 4: Population structure and natural selection analysis

Timing: 10 min

To understand the fine-scale genetic structure of a specie, population structure analysis based on SVs is conducted. Natural selection analysis facilitates comprehension of adaptation to the environment in a population or specie. Here, we only use F_{ST} statistics to evaluate the population differentiation between populations.

14. Conduct the principal component analysis (PCA).

```
# Create a file including reference allele

# VCF_file = population.graphtyper2.aggregated.PASS.GQ13.DP1.GENO0.5.rmRepeat.vcf.gz

>zcat VCF_file | perl -alne 'if(/^#/){next;}print "$F[2]\t$F[3]"' > VCF_ref
```

```
# In the output VCF file of graphtyper2, ALT allele of some SVs contain character ''+'', remove this character

>zcat VCF_file | perl -alne 'if(/^#/){print "$_";next;}$F[4]=~s/\+//;$out=join("\t",@F); print "$out"' | bgzip -c > VCF_file_changeALT.vcf.gz && tabix -p vcf -f VCF_file_changeALT.vcf.gz

# Convert VCF format to PLINK format

>plink1.9 --vcf VCF_file_changeALT --a2-allele VCF_ref --allow-extra-chr --make-bed --out plink_prefix

# Conduct PC analysis using plink software

>plink1.9 --bfile plink_prefix --pca 150 --allow-extra-chr --keep-allele-order --out plink_prefix
```

15. Evaluate population differentiation between two or more populations using F_{ST} statistics (Figure 2).

```
# Conduct Fst analysis using plink software

# The format of cluster_file: <FID><IID><population>, tab-separated

>plink2 --bfile plink_prefix --fst CATPHENO method='wc' report-variants --allow-extra-chr --keep-allele-order --within cluster_file --out plink_prefix
```

Part 5: Differential gene expression analysis

Timing: 1 day (4 threads, variable depending on the number of threads and samples used)

High-differentiated SVs between populations that locates in the CDS region of a gene may have an important functional influence on the development of a species. Differential gene expression (DGE) analysis will help us to better understand the specific function of genes contributing to local adaptation at the transcript level and tissue level.

16. Prepare pair-end RNA sequencing data as the input file of our protocol "SVprotocol_part5.sh".

```
# Part 5: The path of input data to be used in gene expression analysis

#****/ The directory of input and output data to be used

#** In your_gene_expression_analysis_dir, you must have a subdirectory named as "fastq/<sample_name>" including two FASTQ file (<sample_name>_1.fastq.gz and <sample_name>_2.fastq.gz) for pair-end sequencing data

#****/

your_gene_expression_analysis_dir=/your_gene_expression_path/gene_express

# In the config file, we have added the above information. So, in your_gene_expression_analysis_dir, the user must make a subdirectory named as "fastq/<sample_name>" including two FASTQ file (<sample_name>_1.fastq.gz and <sample_name>_2.fastq.gz). For example, for sample SRR1187017, we required two files in the following directory

$your_gene_expression_analysis_dir/fastq/SRR1187017/SRR1187017_1.fastq.gz

$your_gene_expression_analysis_dir/fastq/SRR1187017/SRR1187017_2.fastq.gz
```

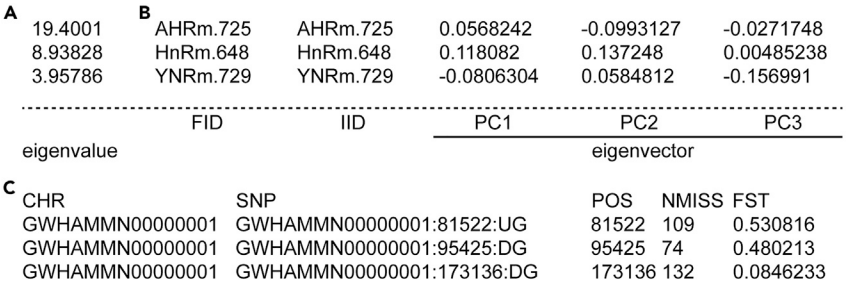


Figure 2. Screenshot of the output files from the population structure and F_{ST} analysis using PLINK software

(A) The eigenvalue file.

(B) The eigenvector file. The meaning of each column is given below the dashed line. FID: family identifier; IID, individual identifier; PC: principal component.

(C) The result of F_{ST} analysis between three populations in *R. microplus*. Identifier "SNP" is the default output of PLINK software.

17. Quality control (QC) for reads must be done before conducting RNA read mapping. FastQC and TrimGalore are widely used and effective software to check read quality, filter out the low-quality reads and remove adaptors. Here, we just give an example of how to run the above two software.

```
# How to run FastQC

>fastqc -q -t 4 -o <OUTDIR> <sample.fastq.gz>

# How to run TrimGalore

>trim_galore --output_dir <OUTDIR> --paired --length <Read_length_cutoff> --quality <Read_
quality_cutoff> --fastqc --gzip --cores <thread> --path_to_cutadapt <cutadapt_software_path>
--basename <outfile_prefix> <SAMPLE_1.fastq.gz> <SAMPLE_2.fastq.gz>
```

18. Map short-read sequence to the reference genome using HISAT2.
19. The R packages (dplyr, edgeR) need to be installed by the user before running the pipeline, which is used to counts the number of reads in each gene region using HTSeq (Figures 4A and 4B).

```
# We have generated a pipeline including all processes (steps 18–19) to calculate read count
for each gene

# Please conducting quality control and filtering before running this pipeline

# Please read the usage before you run this pipeline

# Use the above config file following the argument ''--config''

>sh /this_protocol_script_path/SVprotocol_part5.sh --samplelist SRR_Acc_List.multipleTissue
--config /this_protocol_script_path/example.config --thread 3 --project multipleTissue
```

20. Employ differential gene expression analysis with no replicates using GFOLD (Figure 4C).

```
# After executing the above pipeline, a GFOLD input file with suffix ''gfold.input'' will be
generated in the subdirectory ''140.gfold.<project>'' for each sample

# Choose two samples you want to compare in the GFOLD analysis

>gfold diff -s1 <sample1>.gfold.input -s2 <sample2>.gfold.input -o /outdir/<sample1>_
<sample2>.gfold.diff
```

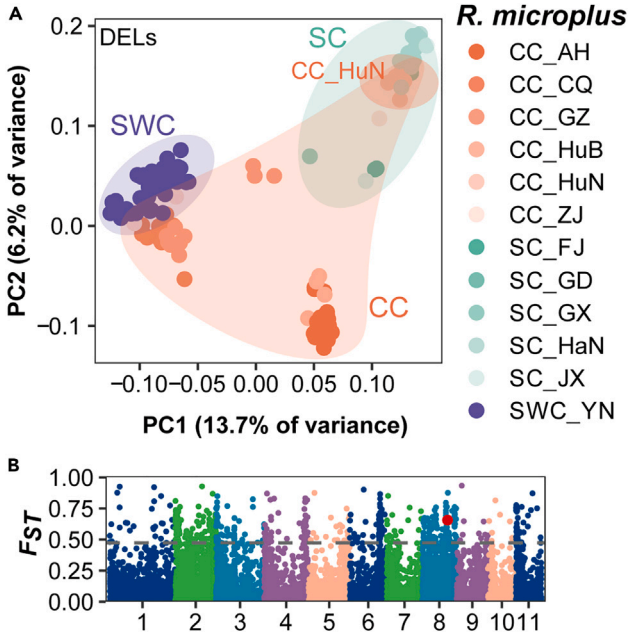


Figure 3. Principal component analysis (PCA) plot and whole genome scan with F_{ST} statistic values

(A) Principal component analysis (PCA) plot of three populations in *R. microplus*. SC: South China; SWC: Southwest China; CC: Central China. The figure is adapted from our previous research article.[23]

(B) Manhattan plot of F_{ST} statistic values. The x-axis represents the chromosome in *R. microplus*. The figure is adapted from our previous research article.[23]

EXPECTED OUTCOMES

Structure variants call sets (part 1 and 2)

The first two parts of our protocol generate one VCF file, containing variant call sets of SVs with recommendatory filtering. The outcome is a file named "population.graphtyper2.aggregated.PASS.GQ13.DP1.GENO0.5.rmRepeat.vcf.gz" in the subdirectory named "140.sv.genotyping".

SV annotation files (part 3)

In this part, a file including gene symbols and gene feature types (CDS, promoter, intronic, intergenic, etc.) that overlapped with SVs is generated. The outcome is a "*.bed" file (Figure 1) and can be used in further natural selection analysis.

Estimated principal components and value of FST statistics for each SV (part 4)

Estimated principal components are generated in this part with two files with suffix "eigenval" and "eigenvec" (Figures 2A and 2B). In addition, the value of F_{ST} statistics between three populations in *R. microplus* is also generated with a file with suffix "fst" (Figure 2C). SV with an empirical *P* value smaller than 0.05 was considered a candidate SV presenting the significant difference between three populations and undergoing positive selection. High-differentiated SVs overlapped with phenotype-related genes might contribute to the local adaptation of phenotype and can be a target gene in further DGE analysis. It is possible to visualize the results using the plotting function in R. Figure 3 depicts a point plot of the samples in *R. microplus* according to the top 2PCs, and a Manhattan plot of F_{ST} values.

Read count for each gene in each sample and input file of GFOLD (part 5)

The read count for each gene in each sample is the first and most important step in the DGE analysis. The outcome is a matrix file with suffix "htseq.gene.count.mat" which contains the gene name and read count for each sample, and a series of files with suffix "gfold.input" that serve as the input file for

A

Gene	Symbol	SRR1187017	SRR1187010	SRR1187007	SRR1187013	SRR1187012	SRR1187005	SRR1186998
Rmic27761.gene	Rmic27761	0	0	0	0	0	0	0
Rmic22179.gene	Rmic22179	0	0	0	0	0	0	0
Rmic04302.gene	Rmic04302	0	0	0	0	7	0	0
Rmic20552.gene	Rmic20552	2	17	0	0	0	0	0
Rmic11379.gene	Rmic11379	0	1	0	0	0	0	0
Rmic08167.gene	Rmic08167	0	0	0	0	0	0	0
Rmic23164.gene	Rmic23164	1	0	0	0	1	0	0
Rmic18247.gene	Rmic18247	6	96	9	9	25	7	4

B

Gene	Symbol	Read_count	Sum_of_exon_length	FPKM
gene-LOC119164228	LOC119164228	0	4575	0
gene-LOC119178103	LOC119178103	153	1425	195.5570024
gene-LOC119177061	LOC119177061	119	1551	139.7436134
gene-LOC119163619	LOC119163619	4	5967	1.220958095
gene-LOC119167605	LOC119167605	0	1656	0

C

```
# This file is generated by gfold V1.1.4 on Mon Aug  8 23:22:26 2022
# Normalization constants :
#    SRR1186998.gfold.input      1908398         1
#    SRR1187005.gfold.input      3300883         2.76543
# The GFOLD value could be considered as a reliable log2 fold change.
# It is positive/negative if the gene is up/down regulated.
# A gene with zero GFOLD value should never be considered as
# differentially expressed. For a comprehensive description of
# GFOLD, please refer to the manual.
```

#GeneSymbol	GeneName	GFOLD(0.01)	E-FDR	log2fdc	1stRPKM	2ndRPKM	
gene-LOC119164228	LOC119164228	0	1	-3.05247	0.229071	0	
gene-LOC119178103	LOC119178103	-0.0970644	1	-0.246803	268.803	362.476	
gene-LOC119177061	LOC119177061	0.302634	1	0.475162	155.747	346.702	
gene-LOC119163619	LOC119163619	0	1	-2.3155	0.70253	0.203083	

Figure 4. Screenshot of the output files produced by the analysis of the differential gene expression

(A) Read count of genes in each sample. The first two columns represent gene information. The retain columns display the read count for each gene in each sample.

(B) A input file for one sample used in GFOLD analysis. The first two columns contain gene information. The retain column denotes the read count of the gene in a sample, the total length of the exon, and the Fragments Per Kilobase of the exon model per Million mapped fragments (FPKM) value, respectively.

(C) The output file of GFOLD analysis. A gene with a positive or negative GFOLD value (|cutoff| = 0.01) could be considered up or down regulated.

GFOLD (Figure 4). Differential gene expression analysis with no replicates was conducted by GFOLD software.

LIMITATIONS

This protocol provides a comprehensive strategy to construct the SVs map from low-coverage sequencing data and explore local adaptation to various environments in any species. However, we are not able to incorporate all relevant methods and algorithms in this protocol, and we only construct some pipelines to run a series of commonly used software and analysis. Considering the diversity of data properties and published software with different efficiencies, the users should make their own decisions regarding software and parameters, such as algorithms for SV detection and genotyping, software for estimating principal components, and software used in DGE analysis.

TROUBLESHOOTING

Problem 1

Manta or Lumpy fails with an error message: "SyntaxError: invalid syntax" (step 2 in part 1: SV detection from the short-read sequences with low coverage).

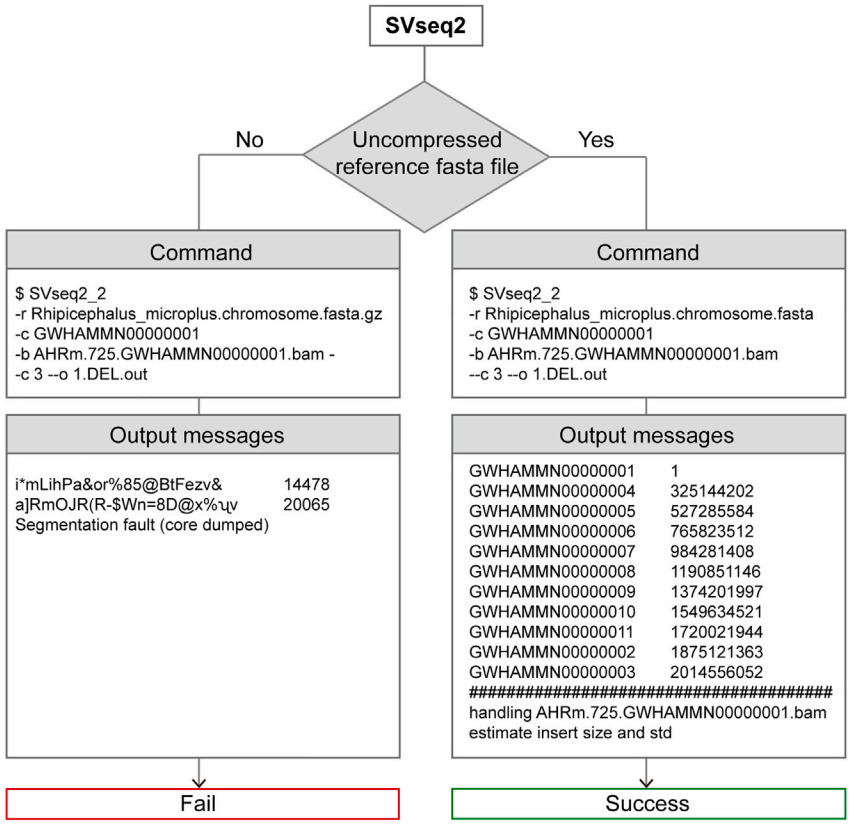


Figure 5. Screenshot of the output message of SVseq2 using different formats of reference genome file

Potential solution

Use python 2+ to run the software.

Problem 2

SVseq2 fails with an error message and ends with "Handling type I reads" (step 2 in part 1: SV detection from the short-read sequences with low coverage).

Potential solution

Skip alignments with MAPQ smaller than 20. You can use the SAMtools with command: samtools view –bam –min-MQ 20 –output out.bam input.bam chromosome. This process is already built into our pipeline.

Problem 3

SVseq2 fail with an error message in some chromosome: "check insert and std" (step 2 in part 1: SV detection from the short-read sequences with low coverage).

Potential solution

Remove the first 300 bp region of the chromosome. If the position of the mapped read starts from 1–150 bp, SVseq2 fails with the above error message. Considering the lower mapping quality at the first end of a chromosome and the generally read length (100–150 bp), we remove the first 300 bp, the cumulative length of two or three reads. You can use the SAMtools with the command: samtools view –bam –out out.bam input.bam chromosome:300. This process is already built into our pipeline. The user can choose another threshold and modify our pipeline.

Problem 4

SVseq2 fail without any error message when you use a compressed reference genome file (Figure 5) (step 5 in part 1: SV detection from the short-read sequences with low coverage).

Potential solution

Input the FASTA file in an uncompressed form with the suffix "fasta".

Problem 5

RepeatModeler fails without any error message when you use a reference genome file including all chromosomes (step 11 in Part 2: SV genotyping and Filtering).

Potential solution

Execute the RepeatModeler for each single chromosome individually.

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Yan Lu (lueyan@fudan.edu.cn).

Materials availability

This study did not generate new materials.

Data and code availability

The datasets analyzed in this study are available in the NGDC database: PRJCA002242. The script generated during this study is available at GitHub: https://github.com/Shuhua-Group/SVanalysis_STARProtocols, or Zenodo: https://zenodo.org/record/7794021#.ZCqB_exByIE, or our laboratory's website: https://pog.fudan.edu.cn/#/software.

ACKNOWLEDGMENTS

This study was supported by the National Natural Science Foundation of China (NSFC) (32030020 and 32288101), the UK Royal Society-Newton Advanced Fellowship (NAF\R1\191094), and the Shanghai Municipal Science and Technology Major Project (grant numbers 2017SHZDZX01). The funders had no role in the study design, data collection, analysis, decision to publish, or preparation of the manuscript.

AUTHOR CONTRIBUTIONS

Y.L. and S.X. conceived and initiated this study. Q.L., B.X., and Y.G. prepared scripts in each part. Q.L. constructed the pipeline and drafted the manuscript. Y.L. and S.X. revised the manuscript. All authors discussed the results and implications and commented on the manuscript.

DECLARATION OF INTERESTS

The authors declare no competing interests.

REFERENCES

1. Jia, N., Wang, J., Shi, W., Du, L., Sun, Y., Zhan, W., Jiang, J.F., Wang, Q., Zhang, B., Ji, P., et al. (2020). Large-Scale Comparative Analyses of Tick Genomes Elucidate Their Genetic Diversity and Vector Capacities. Cell *182*, 1328–1340.e13. https://doi.org/10.1016/j.cell.2020.07.023.

2. Tirloni, L., Braz, G., Nunes, R.D., Gandara, A.C.P., Vieira, L.R., Assumpcao, T.C., Sabadin, G.A., da Silva, R.M., Guizzo, M.G., Machado, J.A., et al. (2020). A physiologic overview of the organ-specific transcriptome of the cattle tick Rhipicephalus microplus. Sci. Rep. *10*, 18296. https://doi.org/10.1038/s41598-020-75341-w.

3. Li, H., and Durbin, R. (2009). Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics *25*, 1754–1760. https://doi.org/10.1093/bioinformatics/btp324.

4. McKenna, A., Hanna, M., Banks, E., Sivachenko, A., Cibulskis, K., Kernytsky, A., Garimella, K., Altshuler, D., Gabriel, S., Daly, M., and DePristo, M.A. (2010). The Genome Analysis Toolkit: A MapReduce Framework for Analyzing Next-Generation DNA Sequencing Data. Genome Res. *20*, 1297–1303.

5. Chen, X., Schulz-Trieglaff, O., Shaw, R., Barnes, B., Schlesinger, F., Källberg, M., Cox, A.J., Kruglyak, S., and Saunders, C.T. (2016). Manta: rapid detection of structural variants and indels for germline and cancer sequencing applications. Bioinformatics *32*, 1220–1222.

https://doi.org/10.1093/bioinformatics/btv710.

6. Layer, R.M., Chiang, C., Quinlan, A.R., and Hall, I.M. (2014). LUMPY: a probabilistic framework for structural variant discovery. Genome Biol. *15*, R84. https://doi.org/10.1186/gb-2014-15-6-r84.

7. Zhang, J., Wang, J., and Wu, Y. (2012). An improved approach for accurate and efficient calling of structural variations with low-coverage sequence data. BMC Bioinf. *13*, S6. https://doi.org/10.1186/1471-2105-13-S6-S6.

8. Eggertsson, H.P., Kristmundsdottir, S., Beyter, D., Jonsson, H., Skuladottir, A., Hardarson, M.T., Gudbjartsson, D.F., Stefansson, K., Halldorsson, B.V., and Melsted, P. (2019). GraphTyper2 enables population-scale genotyping of structural variation using pangenome graphs. Nat. Commun. *10*, 5402. https://doi.org/10.1038/s41467-019-13341-9.

9. Garrison, E., Kronenberg, Z.N., Dawson, E.T., Pedersen, B.S., and Prins, P. (2022). A spectrum of free software tools for processing the VCF variant call format: vcflib, bio-vcf, cyvcf2, hts-nim and slivar. PLoS Comput. Biol. *18*, e1009123. https://doi.org/10.1371/journal.pcbi.1009123.

10. Tarailo-Graovac, M., and Chen, N. (2009). Using RepeatMasker to identify repetitive elements in genomic sequences. Curr. Protoc. Bioinformatics *Chapter 4*, 4.10.1–4.10.14. Chapter 4, Unit 4 10. https://doi.org/10.1002/0471250953.bi0410s05.

11. Numanagic, I., Gökkaya, A.S., Zhang, L., Berger, B., Alkan, C., and Hach, F. (2018). Fast characterization of segmental duplications in genome assemblies. Bioinformatics *34*, i706–i714. https://doi.org/10.1093/bioinformatics/bty586.

12. Quinlan, A.R., and Hall, I.M. (2010). BEDTools: a flexible suite of utilities for comparing genomic features. Bioinformatics *26*, 841–842. https://doi.org/10.1093/bioinformatics/btq033.

13. Danecek, P., Bonfield, J.K., Liddle, J., Marshall, J., Ohan, V., Pollard, M.O., Whitwham, A., Keane, T., McCarthy, S.A., Davies, R.M., and Li, H. (2021). Twelve years of SAMtools and BCFtools. GigaScience *10*, giab008. https://doi.org/10.1093/gigascience/giab008.

14. Danecek, P., Auton, A., Abecasis, G., Albers, C.A., Banks, E., DePristo, M.A., Handsaker, R.E., Lunter, G., Marth, G.T., Sherry, S.T., et al. (2011). The variant call format and VCFtools. Bioinformatics *27*, 2156–2158. https://doi.org/10.1093/bioinformatics/btr330.

15. Chang, C.C., Chow, C.C., Tellier, L.C., Vattikuti, S., Purcell, S.M., and Lee, J.J. (2015). Second-generation PLINK: rising to the challenge of larger and richer datasets. GigaScience *4*, 7. https://doi.org/10.1186/s13742-015-0047-8.

16. Kim, D., Paggi, J.M., Park, C., Bennett, C., and Salzberg, S.L. (2019). Graph-based genome alignment and genotyping with HISAT2 and HISAT-genotype. Nat. Biotechnol. *37*, 907–915. https://doi.org/10.1038/s41587-019-0201-4.

17. Anders, S., Pyl, P.T., and Huber, W. (2015). HTSeq–a Python framework to work with high-throughput sequencing data. Bioinformatics *31*, 166–169. https://doi.org/10.1093/bioinformatics/btu638.

18. Wickham, H., François, R., Henry, L., Müller, K., and Vaughan, D. (2023). Dplyr: A Grammar of Data Manipulation.

19. Robinson, M.D., McCarthy, D.J., and Smyth, G.K. (2010). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics *26*, 139–140. https://doi.org/10.1093/bioinformatics/btp616.

20. Feng, J., Meyer, C.A., Wang, Q., Liu, J.S., Shirley Liu, X., and Zhang, Y. (2012). GFOLD: a generalized fold change for ranking differentially expressed genes from RNA-seq data. Bioinformatics *28*, 2782–2788. https://doi.org/10.1093/bioinformatics/bts515.

21. Kosugi, S., Momozawa, Y., Liu, X., Terao, C., Kubo, M., and Kamatani, Y. (2019). Comprehensive evaluation of structural variation detection algorithms for whole genome sequencing. Genome Biol. *20*, 117. https://doi.org/10.1186/s13059-019-1720-5.

22. Wang, Y., Ling, Y., Gong, J., Zhao, X., Zhou, H., Xie, B., Lou, H., Zhuang, X., Jin, L., et al.; Han100K Initiative (2023). PGG.SV: a whole-genome-sequencing-based structural variant resource and data analysis platform. Nucleic Acids Res. *51*, D1109–D1116. https://doi.org/10.1093/nar/gkac905.

23. Liu, Q., Yang, K., Xie, B., Gao, Y., Xu, S., and Lu, Y. (2023). Mapping structural variations in *Haemaphysalis longicornis* and *Rhipicephalus microplus* reveals vector–pathogen adaptation. iScience *26*, 106398. https://doi.org/10.1016/j.isci.2023.106398.