User Manual of ERVcaller v1.4

February 21, 2022

Citation: Chen X, Li D. ERVcaller: Identifying polymorphic endogenous retrovirus and other transposable element insertions using whole-genome sequencing data. Bioinformatics. 2019 Oct 15;35(20):3913-3922. PMID: 30895294.

Download: https://dllab.org/software/ERVcaller.html

Recent major updates

1) Further increased the accuracy
2) Added the Phred-scale genotype quality and likelihoods
3) Speed up the genotype process significantly
4) Added the function to distinguish missing and none TE insertion genotypes in the combined VCF file for population genomics studies
5) Corrected multiple bugs

Full update log

 # Updates (v1.4):
02/21/2022: Added new Q/A
04/23/2019: Corrected a bug of the genotyping function with the input of a list of BAM files
03/12/2019: Corrected a bug caused by the sample IDs containing the "AS" character
02/15/2019: Re-designed the engineer process to increase the (genotyping) speed significantly
02/10/2019: Added the scripts to distinguish missing genotypes and none TE insertions genotypes for all samples in the combined VCF file
02/06/2019: Corrected the output coordinates of TE insertions with TSD
02/02/2019: Further standardized the VCF format for the usage of bcftools
02/01/2019: Added Phred-scale genotype quality and likelihoods
01/29/2019: Adjusted reciprocal-aligned reference genomic region length using the estimated insert size and SD, which significantly reduced false-positives
01/29/2019: Added a function to estimate insert size and its standard deviation (SD)
01/38/2019: Corrected multiple bugs in the main Perl script
01/24/2019: Corrected a bug in the script combing VCF files from multiple samples
#
 # Updates (v1.3):
11/20/2018: Added the scripts to merge various samples into a list of known TE loci or TE loci detected from the analyzed samples
11/12/2018: Updated the Output in VCF_v4.2 format
11/05/2018: Debugged the support of the BAM files generated by Bowtie2
#
 # Updates (v1.2):
11/01/2018: Further optimized the speed of validation steps
10/21/2018: Supported multiple bam files as the input
10/10/2018: Optimized the validation steps to increase the specificity
#
 # Updates (v1.1):
09/02/2018: Optimized the validation steps to significantly increase the speed
08/28/2018: Updated the parameter of -S to specify the length of split reads used (20 bp by default; >=40 bp is recommended for reads of 150 bp in length)
08/10/2018: Added component to support BAM files using different chromosome IDs as the reference genome, such as "Chr1", "chr1", "1", and "NC_000001.11"
07/17/2018: Corrected bugs for checking input files;
07/17/2018: Corrected the errors for detecting and genotyping TE insertions using single-end sequencing data;
07/16/2018: Re-formatted the output files
07/15/2018: Released ERVcaller Version 1.1 and software manual
#
 # Release (v1.0):

05/27/2018: Released ERVcaller Version 1.0 (a testing version) and software manual

1 Introduction

ERVcaller is a tool designed to accurately detect and genotype non-reference unfixed endogenous retroviruses (ERVs) and other transposon elements (TEs) in the human genome using next-generation sequencing (NGS) data. We evaluated the tool using both simulated and benchmark whole-genome sequencing (WGS) datasets. ERVcaller is capable of accurately detecting various TE insertions of any length, particularly ERVs. It can be applied to both paired-end and single-end WGS, WES, or targeted DNA sequencing data. It supports the use of FASTQ or BAM files(s) generated by different aligners (only BWA, Bowtie were tested). In addition, ERVcaller is capable of detecting insertion breakpoints at single-nucleotide resolution. It allows for the use of either consensus TE sequences or a TE library containing abundant TE sequences as the reference, such as the entire RepBase database. Thus, ERVcaller can be used to detect insertions from highly mutated or novel TE sequences. It is easy to install and use with the command line.

Complementary to ERVcaller, other bioinformatics tools designed to detect large deletions may be used to detect TEs that are present in the human reference genome but not in testing samples.

2 Installation

2.1 Extract the latest ERVcaller installer

$ *tar vxzf ERVcaller_v.1.4.tar.gz*

2.2 Installing dependent software

Users need to successfully install the following software separately and make them available in the default search path (such as by using the Linux command "export" or adding them to your .bashrc).

- BWA-0.7.10: http://bio-bwa.sourceforge.net/bwa.shtml
- Bowtie2: http://bowtie-bio.sourceforge.net/bowtie2/manual.shtml
- SAMtools-1.6 (or later than 1.2): http://www.htslib.org/doc/samtools.html
- R-3.3.2 (or higher): https://www.r-project.org/
- SE_MEI (Modified version included in the Scripts folder of the ERVcaller installer)

2.3 Preparing the references

2.3.1 Human reference genome (hg38 by default. If BAM file(s) are used as input, the same build as the reference used for alignment should be used)

$ *wget http://hgdownload.soe.ucsc.edu/goldenPath/hg38/bigZips/hg38.fa.gz*
$ *gunzip hg38.fa.gz*
$ *bwa index hg38.fa*

2.3.2 TE reference genome. A TE reference is provided by the ERVcaller installer (i.e., the TE consensus sequences consisting of one Alu, LINE1, SVA, and HERV-K consensus sequence each; the human TE library containing 23 TE sequences; and the ERV library extracted from the Repbase database); or a user-defined TE reference library.

$ *cd user_installed_full_path/Database/*
$ *bwa index TE_consensus.fa*

3 Running ERVcaller

3.1 make the installed dependent tools available in the default search path

$ *export PATH=$PATH:$home/bwa-master/*
$ *export PATH=$PATH:$home/samtools-1.6/*
$ *export PATH=$PATH:$home/bowtie2-2.2.7/*
$ *export PATH=$PATH:$home/SE-MEI/*
$ *export PATH=$PATH:$home/Hydra-Version-0.5.3/*
$ *export PATH=$PATH:$home/R/*

3.2 Print help page

$ *perl user_installed_full_path/ERVcaller_v1.4.pl*

3.3 ERVcaller: running command line

$ *perl user_installed_path/ERVcaller_v1.4.pl -i sample_ID -f .bam -H hg38.fa -T TE_consensus.fa –S 20 -BWA_MEM –t threads*

3.3.1 Detecting TE insertions using a BAM file as input

$ *perl user_installed_path/ERVcaller_v.1.4.pl -i TE_seq -f .bam -H hg38.fa -T TE_consensus.fa -I folder_of_input_data -O folder_for_output_files -t 12 -S 20 -BWA_MEM*

3.3.2 Detecting TE insertions using paired-end FASTQ file as input

$ *perl user_installed_path/ERVcaller_v.1.4.pl -i TE_seq -f .fq.gz -H hg38.fa -T TE_consensus.fa -I folder_of_input_data -O folder_for_output_files -t 12 -S 20 -BWA_MEM*

3.3.3 Detecting TE insertions using multiple BAM files as input

$ *perl user_installed_path/ERVcaller_v.1.4.pl -i TE_seq -f .list -H hg38.fa -T TE_consensus.fa -I folder_of_input_data -O folder_for_output_files -t 12 -S 20 -BWA_MEM -m*

3.3.4 Detecting and genotyping TE insertions using a BAM file as input

$ *perl user_installed_path/ERVcaller_v.1.4.pl -i TE_seq -f .bam -H hg38.fa -T TE_consensus.fa -I folder_of_input_data -O folder_for_output_files -t 12 -S 20 -BWA_MEM -G*

3.4 Parameters

All available parameters are listed below. The following four parameters are required: input sample ID (-i), file suffix (-f), human reference genome (-H), and TE reference genomes (-T).

Table 1 List of parameters and their meanings

Parameter (Full name)	**Format**	**Description**
-i | input_sampleID	STRING	Sample ID (*required*)
-f | file_suffix	STRING	The suffix of the input data: zipped FASTQ file (i.e., .fq.gz, and fastq.gz), unzipped FASTQ file (i.e., .fq, and fastq), BAM file(s) (i.e., .bam and .list) (***required***). Default: .bam
-H | Human_reference_genome	STRING	The FASTA file of the human reference genome (*required*)

-T | TE_reference_genomes	STRING	The TE library (FASTA) used for screening (*required*)
-I | Input_directory	STRING	The folder of the input data. Default: Not specified (current working directory)
-O | Output_directory	STRING	The folder for the output files. Default: Not specified (current working directory)
-n | number_of_reads	INTEGER	The minimum number of reads support a TE insertion. Default: 3
-d | data_type	STRING	Data type, including WGS, and RNA-seq. Default: WGS
-s | sequencing_type	STRING	Type of sequencing data: paired-end or single-end. Default: paired-end
-l | length_insertsize	FLOAT	Insert size length (bp) (mean value). It will be estimated if it is not specified
-L | L_std_insertsize	FLOAT	Standard deviation of insert size length (bp). It will be estimated if it is not specified
-r | read_len	INTEGER	Read length (bp): 100, 150, or 250 bp. Default:100
-t | threads	INTEGER	The number of threads. Default: 1
-S | Split	INTEGER	The minimum length for split reads. A longer length, such as 40 or 60 bp for 150 bp reads, is recommended with longer read lengths. Default: 20
-m | multiple_BAM	-	If multiple BAM files are used as the input (input bam files need to be indexed). Default: not specified
-B | BAM_MEM	-	If the BAM file is generated by BWA-MEM (it supports other aligners, including BWA aln, Bowtie2, etc.). Default: not specified
-G | Genotype	-	Genotyping function (input bam file need to be indexed). Default: not specified
-h | help	-	Print this help

PS: with –G or –m, the input bam file need to be indexed using SAMtools.

4 Output files

4.1 Output for each sample

The output VCF file (VCFv4.2) will be generated after running. All annotations are listed below:

```
##fileformat=VCFv4.2
##fileDate=2019121
##source=ERVcaller_v.1.4
##reference=file:hg38.fa
##ALT=<ID=INS:MEI:HERVK,Description="HERVK insertion">
##INFO=<ID=TSD,Number=2,Type=String,Description="NUCLEOTIDE,LEN, Nucleotides and length of the Target Site Duplication (NULL for unknown)">
```

##INFO=<ID=INFOR,Number=6,Type=String,Description="NAME,START,END,LEN,DIRECTION,STATUS; NULL for unknown values. Status of detected TE: 0 = Inconsistent direction for the supporting reads; 1 = One breakpoint detected by only chimeric and/or improper reads without split reads; 2 = Only one breakpoint is detected and covered by split reads; 3 = Two breakpoints detected, and both of them are not covered by split reads; 4 = Two breakpoints detected, and one of them are not covered by split reads; 5 = Two breakpoints detected, and both of them are covered by split reads;">
##INFO=<ID=CR,Number=1,Type=Integer,Description="Number of chimeric and improper reads support the TE insertion">
##INFO=<ID=SR,Number=1,Type=String,Description="Number of split reads support TE insertion and the breakpoint">
##INFO=<ID=GTF,Number=1,Type=String,Description="If the detected TE insertions genotyped">
##INFO=<ID=GR,Number=1,Type=Float,Description="The ratio of the number of reads support TE insertions versus the total number of reads at this TE insertion location">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GQ,Number=1,Type=Float,Description="Genotype quality (Phred transformed)">
##FORMAT=<ID=GL,Number=G,Type=Float,Description="Genotype likelihood">
##FORMAT=<ID=DPI,Number=1,Type=Integer,Description="The number of reads support TE insertions">
##FORMAT=<ID=DPN,Number=1,Type=Integer,Description="The number of reads support non-TE insertions">
#CHROM POS ID REF ALT QUAL FILTER INFO FORMAT TE_seq
chr1 5617379 . T <INS_MEI:HERV> . .
TSD=NULL,NULL;INFOR=HERVK,1,7831,7831,+,4;CR=64;SR=3;GTF=YES;GR=1.000 GT:GQ:GL:DPN:DPI
1/1:40:0,0,1:0:67

4.2 Merging multiple samples

4.2.1 Create a file containing the sample list

4.2.2 Combine multiple samples *with* providing a list of consensus TE loci

$ *perl user_installed_path/Scripts/Combine_VCF_files.pl -l sample_list -c 1KGP.TE.sites.vcf -o Output_merged.vcf*

4.2.3 Combine multiple samples *without* providing a list of consensus TE loci

$ *perl user_installed_path/Scripts/Combine_VCF_files.pl -l sample_list -o Output_merged.vcf*

4.2.4 Calculate the number of reads support non-insertions at the consensus TE loci per sample (it is recommended to filter out low-quality TE loci from the combined VCF file first before running this script)

$ *perl user_installed_path/Scripts/Calculate_reads_among_nonTE_locations.pl -i Output_merged.vcf -S sampleID -o output.nonTE -b bamFile.bam -s paired-end -l length_insertsize -L std_insertsize -r read_length -t threads*

4.2.5 Distinguish missing genotypes and non-insertion genotypes at the consensus TE loci to get the final genotypes for all samples

$ *cat *.nonTE >nonTE_allsamples*
$ *perl user_installed_path/Scripts/Distinguish_nonTE_from_missing_genotype.pl -n nonTE_allsamples -v Output_merged.vcf -o Output_merged-final.vcf*

5 FAQ

5.1 How to install dependent tools

You can follow the links listed below for information about downloading and/or installing all the dependent tools except the modified SE_MEI which is already included with ERVcaller:

- BWA-0.7.10: http://bio-bwa.sourceforge.net/bwa.shtml
- Bowtie2: http://bowtie-bio.sourceforge.net/bowtie2/manual.shtml

- SAMtools-1.6 (or later than 1.2): http://www.htslib.org/doc/samtools.html
- R: https://www.r-project.org/

5.2 How to set the shell environment variables for the installed dependent tools

You can set temporary variables by using the Linux "export" command line before you run ERVcaller every time, or you can modify the shell profile file (ie. .bashrc) for longtime use. You should run for all tools above, except R which is mostly set when installed. For example:
$ export PATH=$PATH:/home/Tools/samtools/

5.3 Where to get the human reference genome

You can download hg38 here: http://hgdownload.soe.ucsc.edu/goldenPath/hg38/bigZips/. It is recommended that the file hg38.fa.gz is downloaded and unzipped for reference indexing.

5.4 Can we use other TE references we collected ourselves?

Yes, you can. You should be able to use any TE reference sequences specific to your research.

5.5 Where can I find test data?

You can find the test input data under the ERVcaller_v.1.4/test/ folder. There is example input data in both BAM and FASTQ format for testing.

There is also an example VCF output file in the folder: ERVcaller_v.1.4/test/example_output/.

5.6 Where can I find more information about the output format?

You can find the full information here: https://samtools.github.io/hts-specs/VCFv4.2.pdf.

5.7 Which parameters were used to produce the example test output file?

The following command line was used to produce the example file:
$ *perl ERVcaller_v.1.4.pl -i TE_seq -f .bam -H hg38.fa -T TE_consensus.fa -G*

5.8 How to speed up ERVcaller

You can use "-t <threads>" to use multi-thread computing. You can skip the genotyping function which can significantly speed up ERVcaller. You may also increase the length of split reads (-S <Split>) to reduce the number of split reads which potentially caused by sequencing errors.

5.9 Do we need to provide the full path to the human reference genome and ERV reference genome in the command line, even if they're in the executable's directory?

Yes.

5.10 Do we need to provide the full path to the ERVcaller in the command line?

Yes.

5.11. How to remove nested TEs (how to detect them in ERVCaller VCF output)?

Because BEDtools is able to identify and remove polymorphic TE insertions (unfixed TEs) that overlap with the same type of reference TE regions (fixed TEs). Users may use

BEDtools “intersect -v” function, together with TE annotation file (annotation can also be obtained from UCSC genome browser), to remove nested TEs.