Fusion-finder: Identify genomic fusion events using directed mapping and machine learning

Kez Cleal, Kate Liddiard, Duncan Baird. 2018

Manual v.1.0.1

Table of Contents

[TLDR 1](#_Toc523907151)

[Installation and dependencies 2](#_Toc523907152)

[Commands 2](#_Toc523907153)

[“run” 2](#_Toc523907154)

[“find-reads” 2](#_Toc523907155)

[“align” 2](#_Toc523907156)

[“call-events” 2](#_Toc523907157)

# TLDR

To install, type in a terminal:

|  |
| --- |
| $ pip install fufi |

Run the pipeline with input bam file and a genome reference in fasta format:

|  |
| --- |
| $ fufi run ref.fa input.bam |

This will produce a .vcf file with structural variants in the same location with the postfix .fufi.vcf

Run using multiple processors, whilst directing output to folder:

|  |
| --- |
| $ fufi run –-procs 12 –dest new\_folder ref.fa input.bam |

‘exclude’ certain regions altogether, whilst limiting the ‘search’ to specified regions, but additionally to ‘limit’ final calls to some other regions of interest:

|  |
| --- |
| $ fufi run –exclude regions1.bed –search regions2.bed –limit regions3.bed ref.fa input.bam |

Explore help messages:

|  |
| --- |
| $ fufi –help  $ fufi run –help |

# Overview

Fusion finder can be used to call structural variants in whole genome sequencing data aligned to a reference genome, although was specifically designed to detect structural variants in amplicons derived from repetitive regions of the genome that have characteristics such as high read-depth and mapping ambiguity.

# Installation and dependencies

To install, type in a terminal:

|  |
| --- |
| $ pip install fufi |

Alternatively download repository from <https://github.com/kcleal> and installed by:

|  |
| --- |
| $ python setup.py install |

Required dependencies:

python >= 2.7 and a c++11 compatible compiler

Python dependencies should be installed automatically but include:

‘click’, ‘numpy’, ‘pandas’, ‘pysam’, ‘quicksect’, ‘pybedtools’, ‘natsort’, ‘networkx’, ‘scikit-learn’

Optional but recommended:

‘bwa mem’ <https://github.com/lh3/bwa> accessible from your path. See the align command for more information.

# Commands

## “run”

Runs the fufi pipeline using default setting for each tool

Options:

|  |  |  |
| --- | --- | --- |
| --include | PATH | .bed file, limit calls to regions. |
| --search | PATH | .bed file, limit search to regions. |
| --exclude | PATH | .bed file, do not search/call SVs within regions. Overrides include/search |
| --clip-length | INTEGER | Minimum soft-clip length; >= threshold are kept. [default: 21] |
| --map-script | PATH | External shell script for mapping. Default is to use bwa mem internally. Script must take positional arguments as: $1 reference genome; $2 .fastq file - must be interleaved if paired-end, otherwise single end reads are assumed; $3 threads to use. |
| -p, --procs | INTEGER RANGE | Processors to use [default: 1], limited to the number of processors available on the machine |
| --dest | PATH | Destination folder to use/create for saving results. Defaults to directory of input bam |
| --help |  | Shows the help message |

Required input arguments:

|  |  |  |
| --- | --- | --- |
| REFERENCE | PATH | Genome reference in fasta format |
| BAM | PATH | The input .bam file to search for structural variants |

Outputs:

Generates a .vcf file of structural variants in either the folder specified by the –dest option, or in the same directory as the input .bam file. The vcf is prost-fixed with .fufi.vcf.

Example:

|  |
| --- |
| $ fufi run –p8 ref.fa input.bam |

## “find-reads”

## “align”

## “call-events”