FAST - FindAmpliconSTructure

FAST is a tool for analyzing the structure and characteristics of amplicons, based on Copy Number Variations and Structural Variations data.

Prerequisites

Python 2.7.x
Packages:

- numpy >= 1.8.2
- pandas == 0.17.0
- xlsxwriter >= 0.6.6
- xlwt >= 0.7.5
- xlrd

Usage

python Fast.py [-h] --config CONFIG [--excel] [--remove_cent]
Parameters:

- --help, -h — show this help message and exit
- --config CONFIG, -c CONFIG — Configuration file
- --excel, -e — generated colored excel file
- --remove_unmappable, -r — remove unmappable areas, specifically centromeres and telomeres data

Inputs

Input files

FAST requires the following input data:

1. Structural Variations (SVs) in the genome of the analyzed, as generated by breakdancer-max[i].
2. Copy Number Variations (CNVs) in the genome of the analyzed sample, as generated by Control-FREEC[ii].

The current version of FAST relies on the file formats of these tools. Thus, it is recommended to use these tools for generating inputs to FAST. However, other tools can be used, as long as the file format is kept.

Configuration File

The configuration file consists of two groups; the first is [PARAMS], in which certain parameters and constants are set. The user may use the default values, or update them. The second group is [FILES], in which the input files are supplied.

Configuration File [PARAMS] parameters:

- WINDOW = 15000
  Window size used for digitization.
- WINDOW_MULT = 3
  The tool calculates threshold as WINDOW_MULT * WINDOW, and uses it for classifying the breakpoints to the corresponding segments.
- CNV_MINIMAL_CN = 6
  Minimal copy number value for considering an area as amplified.
- SV_MINIMAL_SCORE = 90
  Breakpoints with a lower score will be removed from the analysis.
- SV_MINIMAL_NUM_READS = 2
  Breakpoints with a lower number of supporting reads will be removed from the analysis.
- SV_MINIMAL_DELETIONS_REMOVE = 1000
  Deletions which are shorter than this value will be removed from the analysis.

Configuration File [FILES] parameters:

- UNMAPPBLE_INP
  A comma delimited file which contains centromeres and telomeres info. Assumed to contain the following fields:
  chr
  chromStart
  chromEnd
  type (centromere / telomere)
- SAMPLE_ID
- FAST_OUTPUT_DIR
- CNV_INP
  Copy Number Variations (CNVs) file, generated by Control-FREEC
- SV_INP
  Structural Variations (SVs) file, generated by breakdancer-max.

Flow

1. Read the CNV file and SV file and perform some preprocessing to remove some noise:
   a. CNVs file
      - Remove segments with copy number < (configurable) threshold
      - Optional: remove segments in Centromeres
      - Unify consecutive segments
   b. SV file:
      - Remove breakpoints of score less than (configurable) threshold
      - Remove breakpoints which their number of supporting reads is less than (configurable) threshold
      - Remove breakpoints which their type is ‘deletion’ or ‘internal translocation’ and their size is less than (configurable) threshold
      - ‘Digitize’ the start and end points to bins with a (configurable) bin size
      - Unify records which are close to each other and have the same variation type

2. Segment Index and Location Assignment: For each <chr, location> in the SV file, assign the segment it belongs to, and its location in the segment (left/right/middle).
   For the location assignment, the segment is divided into 3 areas: left (first X bases in the segment, X is configurable), right (last X bases in the segment, X is configurable) and middle (the rest). X is defined as WINDOW * WINDOW_MULT, which are 2 parameters in the configuration file. The 'left' and the 'right' are considered as the segment edges.

3. Identify connected segments (i.e. 'Amplicon'): a situation where segment A is connected to segment B (which may be connected to segment C, etc.) as a result of some kind of SV.

4. SV Type Analysis: for each breakpoint in the SV file, determine the structure type of the corresponding amplicon according to the following logic:

   a. **Inverted Duplication (ID)**
      A breakpoint indicating this kind of variation has the following characteristics:
      - Both breakpoint positions belong to the same segment and same location (left / right / middle)
      - Breakpoint type is inversion (INV)

   b. **Tandem Repeat (TR)**
      A breakpoint indicating this kind of variation has the following characteristics:
      - Both breakpoint positions belong to the same segment, but to different locations (e.g. pos1=left of segment 13, pos2=right of segment 13)
      - Breakpoint type is internal translocation (ITX)

   c. **Double Minute (DM)**
      A breakpoint indicating this kind of variation has the following characteristic:
      Each of the breakpoint positions belongs to a different segment

   d. **Other**

5. Amplicon Structure Analysis: the next step is to determine the amplicon structure type. For each amplicon, and for each possible structure type, we sum the supporting reads which are identified with this structure type, according to the previous step. The 'winner' structure type is the one with the majority of supporting reads.

6. Segment Structure Analysis: the analysis of the segment structure type is done similarly to the analysis of the amplicon structure type. Specifically, for each segment, we collect all the breakpoints it belongs to. Then, for each possible structure type, we sum the supporting reads which are identified with this structure type. The 'winner' structure type is the one with the majority of supporting reads.

7. Focused Genes Analysis: as a post-processing step, we can focus on a predefined set of genes of interest. For each gene in the list, we present the corresponding segment (and its structure), and the corresponding amplicon (and its structure).

Output

FAST output is stored in the configured output directory (FAST_OUTPUT_DIR).

The tool outputs the following files:

- <SAMPLE_ID>_cnv_FAST.csv
  CNV file after Fast processing
- <SAMPLE_ID>_sv_FAST.csv
  SV file after Fast processing
- <SAMPLE_ID>_amplicons_FAST.csv
  Amplicon Analysis
- <SAMPLE_ID>_segments_FAST.csv
  Segment Analysis
- <SAMPLE_ID>_focused_genes_FAST.csv
  Analysis of the focused genes loci
- <SAMPLE_ID>_FAST.xlsx
  File summarizing the data of all above files. In this file, colors are assigned to cells, such that each amplicon has its own color. This makes it easier for the human eye to see the relations between the different records.

Test data

Test data of HCC1954 is supplied with the code, under example/data/HCC1954.

Make sure to create FAST_OUTPUT_DIR, as configured in example/data/HCC1954/HCC1954_config.txt

Run by: python Fast.py -c example/data/HCC1954/HCC1954_config.txt -e –r

Author

Michal Devir, michal.devir@gmail.com

To cite please use

<TBD>

[i] We have used breakdancer-max version 1.2.6 (commit 83efb8e)
[ii] We have used Control-FREEC v9.1