BES analysis, alignment with a reference and classification

Created: October 10th, 2019

Updated: October 15th, 2019

For quick help:

```
./bes_analysis.sh -h
```

Introduction

BAC-end sequences (BESs) are paired-end sequences originated from Bacterial Artificial Chromosomes (BACs) and that are separated by a fixed distance. BESs can be aligned to a reference genome and classified according to the orientation between the pairs (Figure 1).

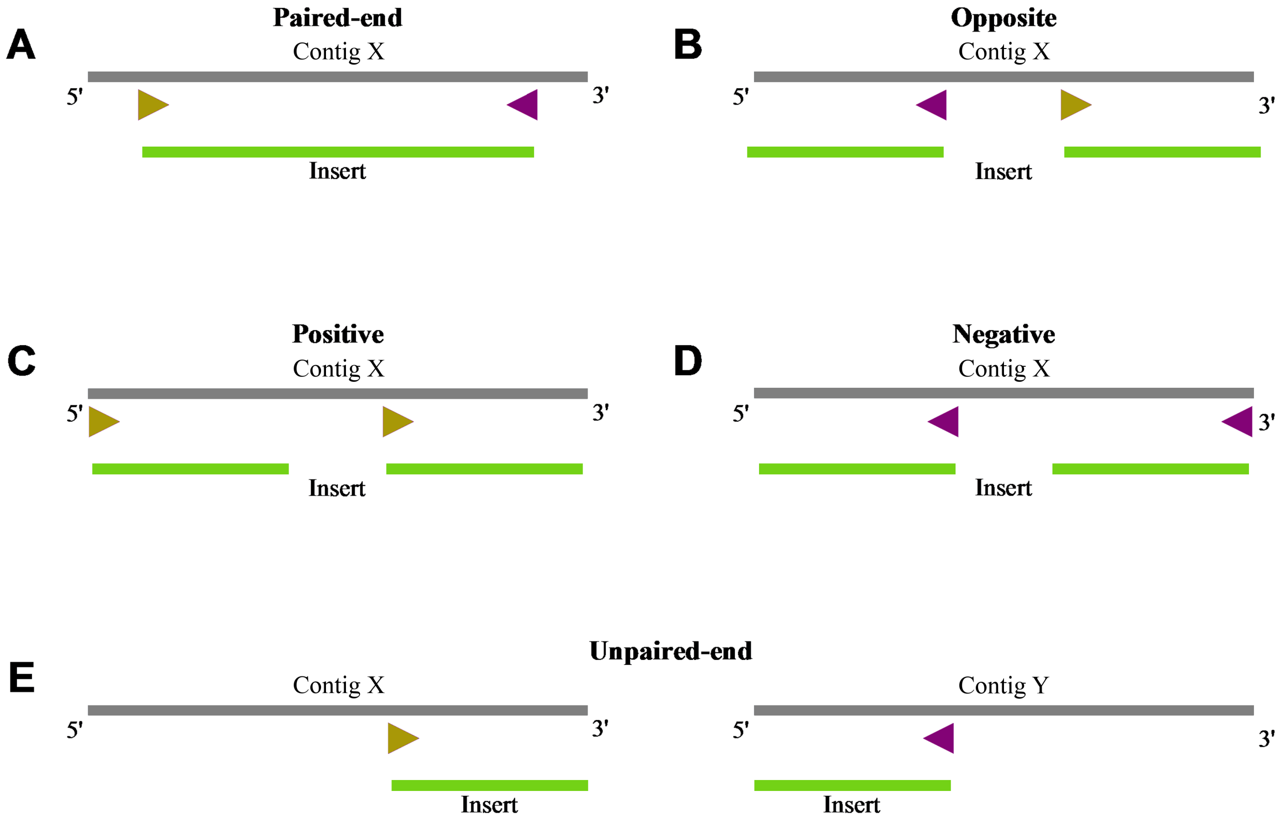


Figure 1. BESs classification according to the alignment orientation between the pairs.

Prerequisites

The following pipeline runs in Linux based platforms. Additionally you should have installed and available in your `$PATH` the following programs:

- **seqret** (from EMBOSS)
- **fastqc**
- **TrimmomaticSE**
- **fastq_to_fasta**
- **blastn**
- **megablast**
- **blat**
- **nucmer**
- **bowtie2**
- **Python** version 3.7.0
- **R** version 3.6.1

The script checks anyway if you have them installed and give generic instructions to install them. The pipeline can be ran without some alignment programs only if they are not called in.

Main Pipeline

The script **bes_analysis.sh** calls in other programs but has the advantage that it automats the process. The script proceeds in five elegible stages:

1. '**abi2fastq**' finds all Sanger reads in **AB1** format of a directory and convert them into **FASTQ** format.
2. '**qual**' evaluates per base quality of the sequences a gives a report using FastQC.
3. '**trim**' trims and filters low quality BESs using Trimmomatic.
4. '**format**' formats the **FASTA** genome headers in order to be correctly read by the script.
5. '**align**' aligns them to a reference genome with one of the following methods:
    - BLAST
    - MEGA (megablast)
    - BLAT
    - NUC (nucmer)
    - BOW (bowtie2)

Convert AB1 files to FASTQ (abi2fastq)

BESs are obtained through Sanger sequencing, therefore, the initial BESs file are in **AB1** format stored in a single directory. To convert such files to **FASTQ** format, a more suitable format for downstreams analysis, one can use the '**abi2fastq**' stage of the **bes_analysis.sh** script giving as input the sequence's directory.

```
./bes_analysis.sh -s abi2fastq -i INPUT.directory
```

The former converts the **AB1** files into **FASTQ** format. In order to preceed with the following analysis, all the **FASTQ** are concatenated by the script. If your sequences are already in **FASTQ** or **FASTA** format, concatenate then using `cat *.fastq > all_bes.fastq` in the command line.

Evaluate sequence quality (qual)

Next step is to analyze the sequences quality. For that, one can run **FASTQC** with a simple command.

```
./bes_analysis.sh -s qual -i INPUT.file
```

Trimming sequences (trim)

This step is perfomerd with Trimmomatic as follows.

```
./bes_analysis.sh -s trim -i INPUT.file
```

Trimmomatic parameters were optimized to generate the sequences LIBGSS_039348; however, additional input parameters can be modified using this aplicaction. These are: `-c HEADCROP`, `-p PHRED_CODING`, `-w WINDOW_SIZE`, `-l LENGTH`, and `-t THRESHOLD`.

After running this stage, you can run again the '**qual**' stage to verify the sequences quality.

Genome FASTA headers formatting (format)

The pipeline requieres that the contigs size information of the reference genome is included in the header for downstream analysis. Therefore, before running the following stages, make sure that your genome FASTA headers have this format: `>sequence1|size1234`. If not, format your reference genome FASTA headers with the following command:

```
./bes_analysis.sh -s format -i none -g GENOME.fasta
```

The input `-i none` is required to perform the argument parsing correctly.

BESs alignmet with a reference genome (align)

This stage is the main function of the script. The objective is to locate the sequences in a reference genome. This can be done using:

```
./bes_analysis.sh -s align -i INPUT.file -g GENOME.fasta -m METHOD
```

The script has the ability to align the BESs sequences (or any other sequences) using one of the five available aligning programs. These are BLAST, MEGABLAST, BLAT, NUCMER, and BOWTIE2. For that, you only need to subsititue method in the `-m` argument for 'BLAST', 'MEGA', 'BLAT', 'NUC' or 'BOW', respectively. For example:

```
./bes_analysis.sh -s align -i INPUT.file -g GENOME.fasta -m BLAST
```

An additional optional argument for this stage is `-e` or E-value for BLAST and MEGABLAST. Defatul E-value is 0.000001.

After the alignmet, the output of the methods that retrieves more than one alignment per sequence goes trough a filtering process in which the longest alignment is selected. In case of a tie, the firts alignment is selected. Then, the script converts any alignment output into **GFF3** format for easier visualization in the Integrative Genomics Viewer (IGV). Therefore, this script is also useful to align any other sequences and visualize them in the IGV. The script performs by default a screening calling in a Python script included in this repository (**screening.py**). This script classifies the BESs pairs according to their orientation in single-end, paired-end, unpaired-end, opposite, positive and negative. For the screening process to work, sequences form the same plasmid (paired-end sequences) must be next to each other in the initial file (for example, sequence_from_plasmid1.fw must be followed by sequence_from_plasmid1.rv, and so on). This screening option can be disabled by using the optional argument `-q no` (default is 'yes').

The results stored are stored in a output directory called 'BES_out' (you can specify your own directory with the input argument `-o OUTPUT_DIR`) and consist of the alingmet method default output, and initial GFF3 file, a final formatted **GFF3**, and a summary table in a plain text file. The final formatted **GFF3** file can aid in visual inpsection in a genome browser, for example the IGV (Figure 2).

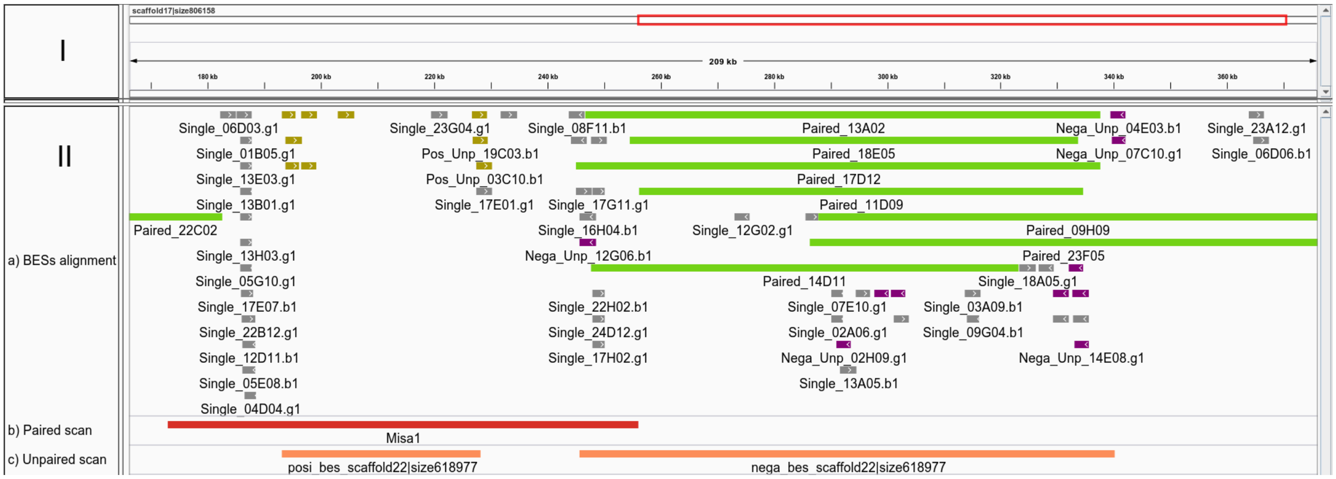


Figure 2. Example vizualization of the formatted GFF3 file. The region of interest (top panel, I) was visualized in the IGV, along with the evidence information (bottom panel, II). a) Formatted GFF3 file of the BESs alignments. Paired-end alignments are shown with its simulated insert in green, unpaired-end BESs aligned positively and negatively are presented in yellow and purple, respectively; single alignmnets are shown in gray. Positive, negative and opposite alignments, although not shown in the image, are assigned the colors pink, cyan and red, respectively. b) Paired-end BESs scanning results (red line). c) Unpaired-end scanning results (orange lines).

Extra analysis

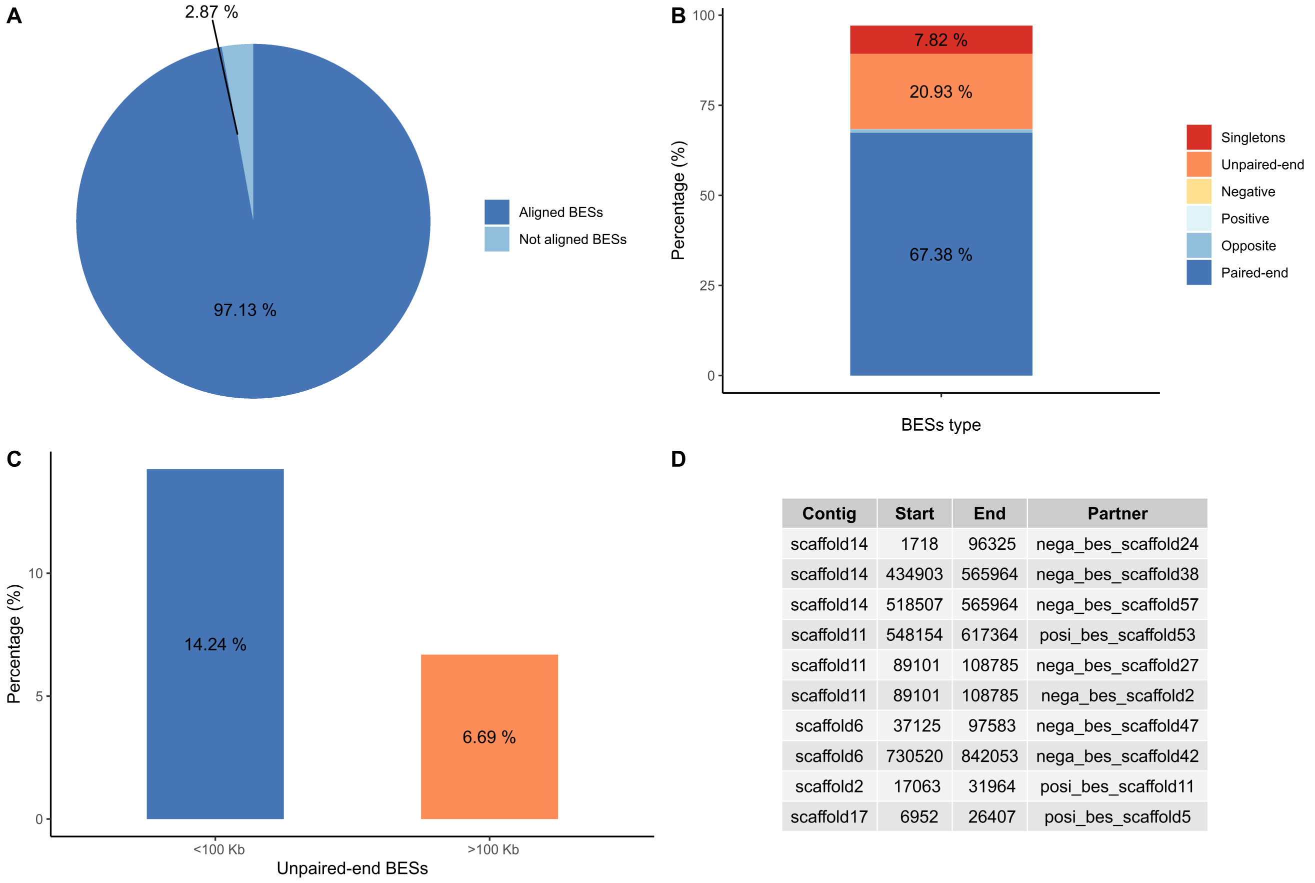
Regions not sustained by paired-end alignment are and indication of a misassembly, or strain specific differences; while region enriched with unpaired-end alignment suggest possible contig joins. Visual inpsection can be time consuming; therefore, two additional scripts that help to detect regions not spanned by paired-end alignments (**sliding_paired.py**) and regions enriched with unpaired-end alignments (**sliding_unpaired.py**) can be ran over the final formatted **GFF3** file. These scripts work using an sliding window algorithm and Pandas dataframes.

```
./sliding_paired.py formatted_GFF_final.gff > lacking_paired_regions.bed
./sliding_unpaired.py formatted_GFF_final.gff > enriched_unpaired_regions.bed
```

The ouput of the former commands are **BED** files that can be loaded simultaneously with the formatted **GFF3** file to aid with visual inspection (Figure 2, panel IIb and IIc).

Finally, you can also run a R script to generate a graphic summary of the results (**graphic_summary.R**) from the BESs alignment summary, the number of input BESs, the enriched_unpaired_regions.bed file, and the output image name (Figure 3). For example:

```
Rscript graphic_summary.R bes_vs_genome_method.summary 4004 enriched_unpaired_re
```



Contig	Start	End	Partner
scaffold14	1718	96325	nega_bes_scaffold24
scaffold14	434903	565964	nega_bes_scaffold38
scaffold14	518507	565964	nega_bes_scaffold57
scaffold11	548154	617364	posi_bes_scaffold53
scaffold11	89101	108785	nega_bes_scaffold27
scaffold11	89101	108785	nega_bes_scaffold2
scaffold6	37125	97583	nega_bes_scaffold47
scaffold6	730520	842053	nega_bes_scaffold42
scaffold2	17063	31964	posi_bes_scaffold11
scaffold17	6952	26407	posi_bes_scaffold5

Figure 3. Graphic example summary. A) Percentage of aligned BESs. B) percentage of the different BESs types. C) percentage of unpaired-end BES according to the insert size. Negative, positive and opposite BESs bars are not observed because they are less than 1%. D) Regions with consistent unpaired-end BESs aligned (only the first ten are shown).

Edit your **BED** file `enriched_unpaired_regions.bed` with `sed` in the reference sequences are too long to be properly diplayed in the graphic summary.