Drosophila Inversion Breakpoints (*DIB*) pipeline manual

DIB (Drosophila Inversion Breakpoints) is an automated pipeline for locating inversion breakpoints when starting with sequence reads. As shown in Figure 1, it is mainly based on the AfterQC, Wgsim, Blastn, and TakeaBreak softwares. If requested, it also uses the European Nucleotide Archive (ENA), to download data SRR files, and the National Center for Biotechnology Information (NCBI), to download the reference genome assembly.

Installation

In order to run DIB, Docker must be installed. You can install Docker on your system by following the comprehensive guidelines available at the Bioinformatics Docker Images Project (https://pegi3s.github.io/dockerfiles/) or by following the Docker manual instructions (https://docs.docker.com/get-started/overview/). Docker is compatible with all major operating systems, including Linux, Windows, and Mac.

Once Docker is installed, the DIB Docker image can be pulled by writing in the command line: “docker pull pegi3s/dib”. Ensure that you have the latest version of the image to benefit from performance enhancements and access the most recent features.

Prerequisites

Before using the pipeline, ensure that you have the following information/data:

SRR Data: Obtain a list of Run Accession Numbers (SRR) from the European Nucleotide Archive (ENA) database. Remember to store this list in the project directory. Alternatively, provide SRR files, in either FASTA or FASTQ format, in a folder named “SRR_data”, located in the project directory.

Reference Genome: Identify the NCBI RefSeq Assembly number corresponding to the relevant reference genome. Alternatively, provide a FASTA file in a folder named “reference_file”, located in the project directory.

Configuration File

Before initiating the pipeline, it is essential to configure the “config” file (see Figure 2). The following parameters must be specified:

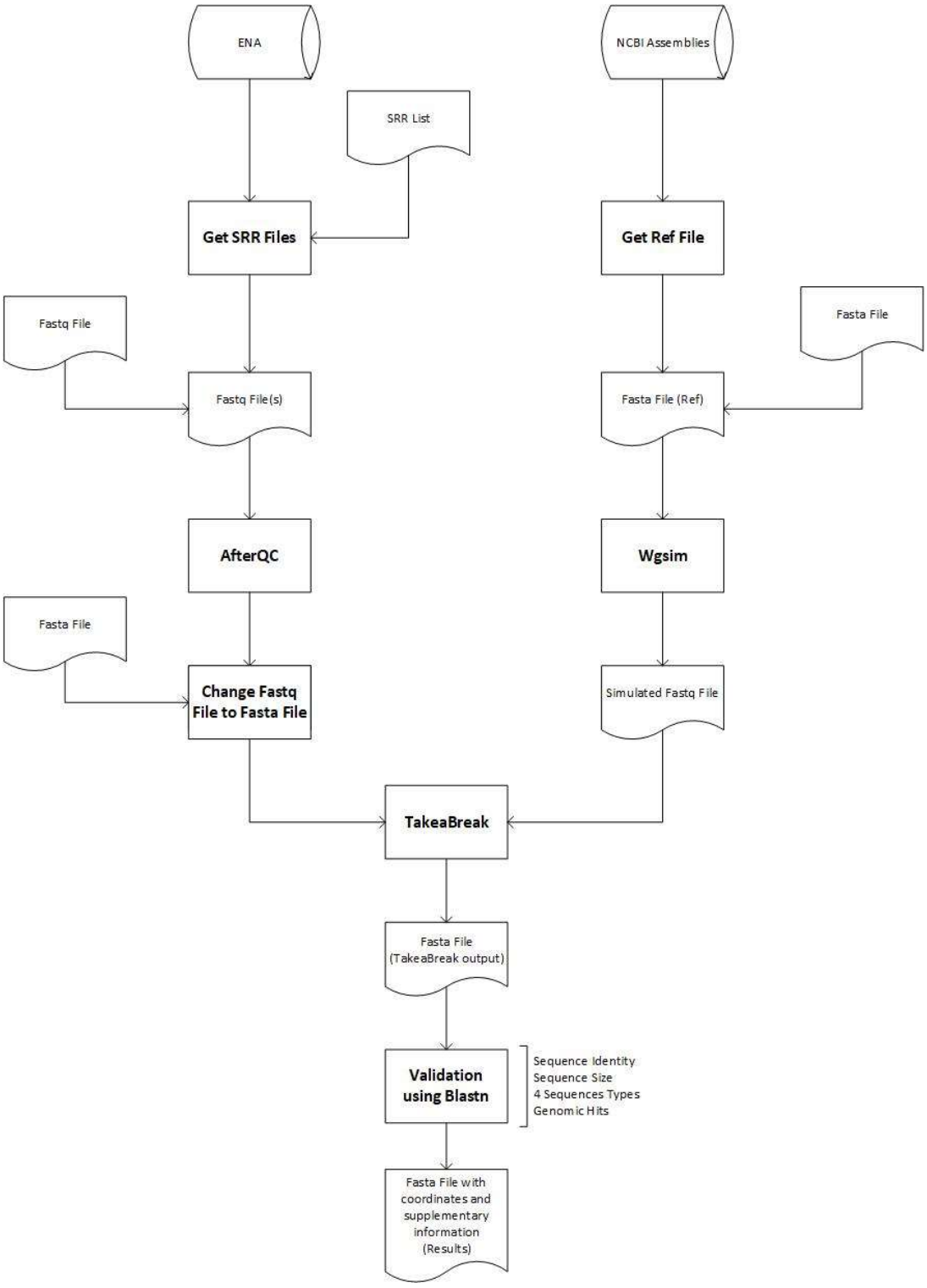

Figure 1. *Drosophila* Inversion Breakpoints (*DIB*) Workflow.

- **dir:** Set the directory path to the project directory.
- **get_reference:** Specify “Y” or “y” if automatic downloading of the reference genome is required; choose “N” or “n” if the reference genome will be provided.
- **reference:** If “Y” or “y” is chosen for (**get_reference**) the NCBI RefSeq Assembly number should be provided for the automatic download of the reference genome. If “N” or “n” is selected for (**get_reference**) the filename of the reference genome should be provided. The reference genome should be stored in a folder named “reference_file” within the project directory.
- **list:** Indicate the name of the file containing the list of SRR numbers corresponding to the data. This list file must be located in the project directory.
- **format:** Choose either “fastq” or “fasta” based on the format of the SRR data.
- **Wgsim options (default values in brackets):**
  - **sequence_size1:** Specifies the length of the first read. [70]
  - **sequence_size2:** Specifies the length of the second read (for paired-end sequencing). [70]
  - **rate_mutations:** Sets the mutation rate. [0.0010]
  - **fraction_indels:** Defines the fraction of indels (insertions/deletions) in the simulated reads. [0.15]
  - **prob_indel:** Sets the probability of an indel happening. [0.30]
  - **error_rate:** Specifies the sequencing error rate. [0.020]
  - **standard_deviation:** Sets the standard deviation of the fragment size. [-1]
  - **read_pairs:** Determines the number of read pairs to generate. [1000000]
  - **seed:** Provides a random seed for reproducibility. [-1]
- **Filtering Options:**
    - **seq_ident:** sequence identity
    - **align_length:** alignment length
    - **gen_hits**: maximum number of genome hits. In Figure 2, these parameters are set to highlight cases where sequence identity between a putative breakpoint sequence and the reference genome exceeds 97%, the alignment length is greater than 50 base pairs, and the number of hits in the reference genome is less than 5. Only cases where four sequence types are identified are considered valid for analysis.

```
# Project Directory
dir=/home/evolution3/Documents/Joao_Cunha/pipeline

# Reference options
get_reference=n
reference=GCF_000001215.4_1200pb.fna

# SRR options
list=SRR_list.txt
format=fastq

# Wgsim options
sequence_size1=300
sequence_size2=300
rate_mutations=0.010
fraction_indels=0.15
prob_indel=0.30
error_rate=0
standard_deviation=50
read_pairs=1000000
seed=0

# Filtering options
seq_ident=97.000
align_length=50
gen_hits=5
```

Figure 2. Example of a Configuration file.

Running the Pipeline

To execute the pipeline, access the terminal and navigate to the project directory. The pipeline will sequentially execute the various steps outlined in the flowchart, including data collection, data processing, inversion breakpoint detection, and result filtering.

To run the pipeline you should adapt and run the following command: **docker run -v /var/run/docker.sock:/var/run/docker.sock -v /your/data/dir:/data pegi3s/dib bash -c "./main"**

Make sure to replace "**/your/data/dir**" with the actual path to your project's directory. This is where the output files (results) generated by the pipeline can be found.

Output Format and Information

Figure 3 illustrates the typical format of a results file. In this example:

```
1 >INV_a-u_0_rep_0 Not found in reference genome
2 CGACTGTATGACATTGACTCTTTTGCCAAAATGTTCCGCTCCATGATCCTTTCTGAAACC
3 >INV_v-b_0_rep_0 Not found in reference genome
4 TCTGAAACCAAATTGATGGGAAGGGATTGCCAGCTGGCCAGGGTGACCCAATTTATTCTA
5 >INV_a-vbar_0_rep_0 NC_004354.4 100.000 60 370 429  Found in reference genome
6 CGACTGTATGACATTGACTCTTTTGCCAAAGCAATCCCTTCCCATCAATTTGGTTTCAGA
7 >INV_ubar-b_0_rep_0 NC_004354.4 100.000 60 421 480  Found in reference genome
8 GGTTTCAGAAAGGATCATGGAGCGGAACATCAGCTGGCCAGGGTGACCCAATTTATTCTA
```

Figure 3. Example of a Results file.

- Every inversion event corresponds to four distinct entries within the FASTA file: The first two entries pertain to the breakpoint sequences, denoted as **"INV_a-u_0"** and **"INV_v-b_0"** which are anticipated to be present in one genome. Subsequently, the final two entries represent the corresponding breakpoint sequences in the other genome, designated as **"INV_a-vbar_0"** and **"INV_ubar-b_0"**. Additionally, the size of the exact repeat detected at the breakpoints is indicated in each entry header (e.g., **"rep_0"**).
- **"NC_004354.4"** is the reference genome where these breakpoints were found.
- **"100.000"** indicates a 100% match or identity between the query and reference sequences.
- **"60"** is the length of the alignment or sequence overlap.
- **"370"** and "**421"** are the start positions of the alignment in the query for the first breakpoint.
- **"429"** and **"480"** are the end positions of the alignment in the query for the second breakpoint.

Note: It should be noted that these results were obtained using simulated inversion data for testing purposes. As you transition to real datasets and more complex research scenarios, the final FASTA file may be more complex.

Generating test data: The script named "simulate_data" can be used to generate sequence read files containing inversions at locations determined by the user. It is made available for testing purposes, for instance, to understand how the pipeline would behave if the breakpoints are located on top of a repetitive element. In order to execute this script, the user should run the following code:

docker run -v /var/run/docker.sock:/var/run/docker.sock -v /your/data/dir:/data pegi3s/dib bash -c "./simulate_data <reference_file> /your/data/dir <first.breakpoint.pos> <second.breakpoint.pos> <options>"

In this command, you should replace:

- **/your/data/dir** to point to the directory that contains the FASTA file you want to analyze (appears twice).
- **<reference_file>** with the name of the input FASTA file you want to use (placed in the project directory).
- **<first.breakpoint.pos>** with the position of the first breakpoint.
- **<second.breakpoint.pos>** with the position of the second breakpoint.
- **<options>** with the specific options of the Wgsim tool. You can substitute this parameter by just the letter "**s**", meaning that you want to use the standard values shown in Figure 2. Otherwise, you need to specify the Wgsim parameters by the following order: **sequence_size1**, **sequence_size2**, **rate_mutations**, **fraction_indels**, **prob_indel**, **error_rate**, **standard_deviation**, **read_pairs**, **seed**.