Figbird User Manual

Version: 0.2.0
Date: 21-03-2022
Made by: Atif Hasan Rahman and Sumit Tarafder

Introduction:

Figbird(**FI**lling **G**aps **by** **I**terative **Read** **D**istribution) software is designed as part of a novel gap filling approach proposed recently. It can successfully fill up gaps in any draft genome assembly consisting of gapped scaffolds using 2nd generation Illumina read sequences. Some main advantages of Figbird are:

1. Supports read pairs of both smaller inserts(~200 bp) and larger inserts (~3500 bp).
2. Utilizes probabilistic methods instead of graph based methods and thus is helpful to eradicate repeat related problems associated with graphs.
3. It is memory efficient compared to other state-of-the-art tools.

Dependencies:

The software can run on Linux and Mac systems with a few dependencies listed below:

1. Bowtie2: Used for mapping read pairs to gapped scaffolds. The default bowtie2 version used and given with the software is 2.2.3(Linux). But user can download their preferred version from: https://github.com/BenLangmead/bowtie2
- If you want to use the given version inside the software, then **unzip the folder and compile it with command "make".**

2. The software is developed in C++ and requires GNU g++(version 4.8 or greater) to compile the codes and the driver script is written in bash and requires GNU bash (version 4.3 or greater)

3. The software also needs GNU uitlity 'bc'[basic calculator]. If you don't have bc in your system, run the following command:

```
      - sudo apt install bc
```

4. A command line JSON processor library 'jq'. You can install jq from the following github page:
      https://stedolan.github.io/jq/

5. [Optional]Python is required only if you want to assess the quality of filled gaps using QUAST software. The exact version of QUAST along with necessary correction files as depicted in paper is already attached with the software. Unzip the folder before using it. There is no need to install QUAST.

Download:

The software can be downloaded from the following github URL:

https://github.com/SumitTarafder/Figbird

The folder “Figbird” contains the followings:
1) Bowtie2 and QUAST source folders in zip format
2) Instruction PDF
3) Multiple .cpp files for Figbird tool and .py files for QUAST
4) A driver bash script named RunFigbird.sh
5) A JSON configuration script named Config.json
6) A compiled linux library file named 'jq' to parse the JSON script.

Run:

Go to the command line and give the following command as input:

```
chmod a+x RunFigbird.sh && ./RunFigbird.sh Config.json
```

It's mandatory to input a configuration file in json format containing all the parameters as instructed below:

Parameter Configuration:

To configure the file paths and other parameters, Figbird uses a configuration file in JSON format. A sample file is included in the software and the users must change the file accordingly maintaining the format.

Following is the list of parameters in the JSON file with explanations:

Draft_genome: Path to the gapped draft genome to fill.

Bowtie2: Path to the bowtie2 executables. If your bowtie2 is in system path, then put “” in the path.

Output_Folder: Path to the directory where all the outputs will be stored.

Reference_Genome: This is optional and only needed if you want to evaluate the quality of the filled assembly using QUAST.

Parameters:

1. **numthreads**: Number of threads used during bowtie2 alignment and gap filling procedure.
2. **evaluation**: Put 1 if you want to assess with QUAST or 0 otherwise.
3. **gaplen_negative_overlap**: We have allowed negative overlap of reads in our method i.e a gap can be diminished if the corresponding left and right flank of the gap merges with supporting reads for verification. Enter the maximum length of the

gaps for which this method will be applicable.[Default: 30]

4. **default**: If you want to fix the order of the reads usage along with their number of iterations, put 0. Otherwise, put 1 for default approach. If you put 1, then information [6-9] for read pairs will not be needed to specify and can be left alone.

5. **trim_len**: Default value has been set to 10. This parameter defines the amount of nucleotides being chopped off from either side of the gapped regions as this is the stopping point for the assemblies and highly likely to contain erroneous sequence.

6. **set_inputmean**: Default value is 0. It can be set to 0 or 1. Users can set this parameter to 1 to set the minimum scaffold length equal to the “avg_insert_size” of the read library to reduce bias towards shorter insert sizes during alignment for learning distributions. Otherwise, set it to 0 for no limits.

Read_Pairs: Input your paired read libraries one by one along with the necessary information:

1. **path_1**: Path to first of the read pair files
2. **path_2**: Path to second of the read pair files
3. **avg_insert_size**: Average insert size of the read pair library.
4. **is_reverse**: If your read pair files are already in forward-reverse(FR) orientation then put 0, otherwise put 1. In case a 1 is given, we will reverse complement both the files of the the input read pair.
5. **max_read_len**: Maximum read length of the library
6. **serial_num**: The order of reads usage for filling gaps
7. **num_itr_partial**: We will use both one end partially aligned and one end unmapped reads for each read pair for gap filling purpose. Enter the itration count for partial approach here.
8. **num_itr_unmapped**: Enter the itration count for unmapped approach here.
9. **order**: Put the order for Which one between partial and unmapped method will be applied first.

* [Users must input atleast one library of read pair files and all 9 required information per library to start gap filling]

Output:

A folder named Figbird will be created in the user given “Output_Folder” directory and following files and folders will be inside this folder.

1) Alignments: A folder that will contain the sam alignments from bowtie2 and our softwares internal formatted alignment files

“myout.sam”

2) Bowtie2_indexFiles: Index files generated by bowtie2 during alignment.

3) QUAST_Results: Contains the six evaluation metrics computed by QUAST in a file named “Result.txt” along with all the other detailed outputs generated by QUAST software.

4) Gaps: Contains two different types of files per gap that stores one end partially mapped reads and one end completely unmapped reads in these two different files.

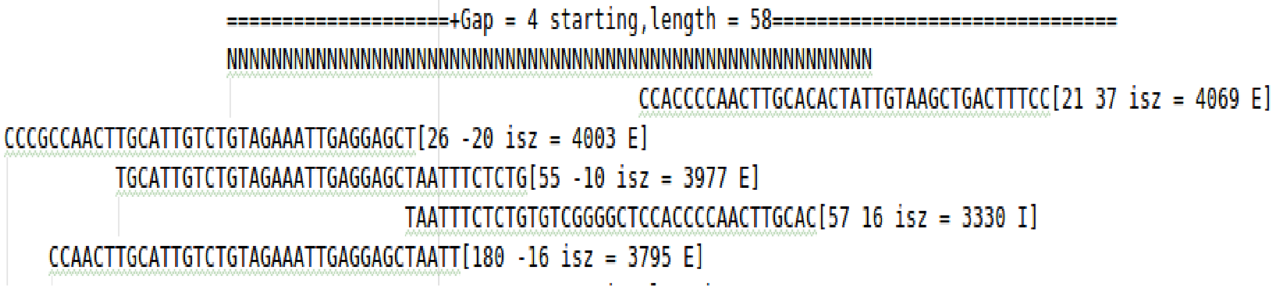
5) Temp: Holds all the intermediate files generated during execution of the script.

6) Filled_Scaffolds: This folder will store:

a) All the gap filled scaffolds per iteration [Intermediate result will be available even though entire execution may not be over]

b) A folder named “Individual_gaps” which contains text files of the format gapout_*.txt and alignment_*.txt per iteration. It will also contain two other files named “combined_gapstring.txt” and “Individual_gaps.txt” which contain the merged sequence of the filled gap over all iterations completed by Figbird tool with details description.

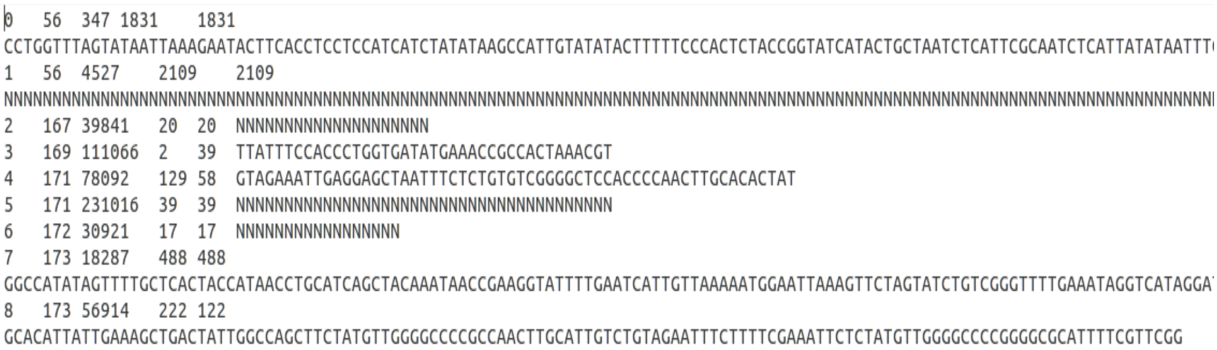
The alignment file is a visual representation of the local assembly of reads per gap.



```
                    ====================+Gap = 4 starting,length = 58===============================
                    NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
                                                         CCACCCCAACTTGCACACTATTGTAAGCTGACTTTCC[21 37 isz = 4069 E]
CCCGCCAACTTGCATTGTCTGTAGAAATTGAGGAGCT[26 -20 isz = 4003 E]
          TGCATTGTCTGTAGAAATTGAGGAGCTAATTTCTCTG[55 -10 isz = 3977 E]
                                    TAATTTCTCTGTGTCGGGGCTCCACCCCAACTTGCAC[57 16 isz = 3330 I]
    CCAACTTGCATTGTCTGTAGAAATTGAGGAGCTAATT[180 -16 isz = 3795 E]
```

Each line of the gapout text file(shown below) contains five columns that shows information about the filled gaps in the following format:

Column-1: Gap number starting from index 0
Column-2: The serial no. of scaffold that the gap resides in
Column-3: Starting position of the gap in the scaffold
Column-4: Given length of the gap in draft assembly file
Column-5: Predicted length of the gap by our tool
Column-6: Predicted gap string



```
0   56  347 1831    1831
CCTGGTTTAGTATAATTAAAGAATACTTCACCTCCTCCATCATCTATATAAGCCATTGTATATACTTTTTCCCACTCTACCGGTATCATACTGCTAATCTCATTCGCAATCTCATTATATATAATTT
1   56  4527    2109    2109
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
2   167 39841   20  20  NNNNNNNNNNNNNNNNNNNN
3   169 111066  2   39  TTATTTCCACCCTGGTGATATGAAACCGCCACTAAACGT
4   171 78092   129 58  GTAGAAATTGAGGAGCTAATTTCTCTGTGTCGGGGCTCCACCCCAACTTGCACACTAT
5   171 231016  39  39  NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
6   172 30921   17  17  NNNNNNNNNNNNNNNNN
7   173 18287   488 488
GGCCATATAGTTTTGCTCACTACCATAACCTGCATCAGCTACAAATAACCGAAGGTATTTTGAATCATTGTTAAAAATGGAATTAAAGTTCTAGTATCTGTCGGGTTTTGAAATAGGTCATAGGA
8   173 56914   222 122
GCACATTATTGAAAGCTGACTATTGGCCAGCTTCTATGTTGGGGCCCCGCCAACTTGCATTGTCTGTAGAATTTCTTTTCGAAATTCTCTATGTTGGGGCCCCGGGGCGCATTTTCGTTCGG
```