

JAST Documentation

DELESTRE Clément

July 17, 2014

Contents

1	About JAST	2
1.1	Dependencies	2
2	How it works	3
2.1	Input file(s)	3
2.2	Stdout and stderr	4
2.3	About the pipeline	4
3	For programmers	5

1 About JAST

JAST stands for **J**ava **A**ssembly and **S**caffolding **T**ool. This program performs assembling and scaffolding from paired-end Illumina library and uses a reference genome.

1.1 Dependencies

JAST uses the following softwares : (see 1)

- Flexbar for filtering reads.
- A5 for assembling reads.
- Bowtie2 to map the filtered reads against the reference genome (a SAM file is required for Colombus).
- Colombus *via* VelvetOptimizer for scaffolding using a reference genome.
- And SSPACE to improve obtained scaffolds.

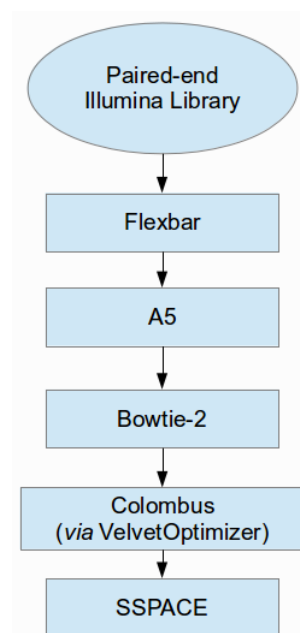


Figure 1: Simple pipeline of JAST.

These programs have to be properly installed on your computer before you can use JAST. As for every Java programs, you also need a JRE to be able to use it. Please note the one used to test JAST was 1.7.

2 How it works

Here we describe how to use JAST. Please note this software doesn't have GUI¹ so we assume you are able to deal with command lines. All options will be not discussed here, to display all of them, use -h or - -help.

2.1 Input file(s)

Illumina library

JAST was designed to work with paired-end Illumina library. Your files name *MUST* follow this format : *myReads_1.fastq* and *myReads_2.fastq*. The first one must be use with the -r or - -reads option and the second one with -p or - -paired.

Config files

If you want to specify some options to softwares used by JAST you can do this with a config file. Each config files must contain each options separated by a break line. VelvetOptimizer is the only software requires a config file. Examples of config files can be found at : <https://github.com/AgResearch/JAST/tree/master/ExampleConfigFiles>

SSPACE library

JAST requires a SSPACE library file ; equivalent to the file you specify with '-l' option when you use SSPACE. Please note that the first line must contains only the 3 last columns : the other information (name and fastq files) will be written in a file nammed with _JAST extension and the library will be nammed JASTlib (see fig. 2). This file must be specified with -l or - -sspacelib option.



Figure 2: Example of SSPACE library file created by JAST (the filename are reds).

The file "library.txt" on github <https://github.com/AgResearch/JAST/tree/master/ExampleConfigFiles> is an example of SSPACE library file required by JAST.

¹Graphical User Interface

2.2 Stdout and stderr

Messages displayed to stdout by JAST during its process, start with [JAST]. When an error appears, messages displayed to stderr start with [JAST_Error]. Messages which do not start with these symbols belong to others softwares.

2.3 About the pipeline

The pipeline is more detailed at fig. 3. On this figure we assume that "?" could be "1" or "2", "n" is an integer, "genome" is the reference genome (- - ref option) and "output" is the output name specified by -o or - - output option.

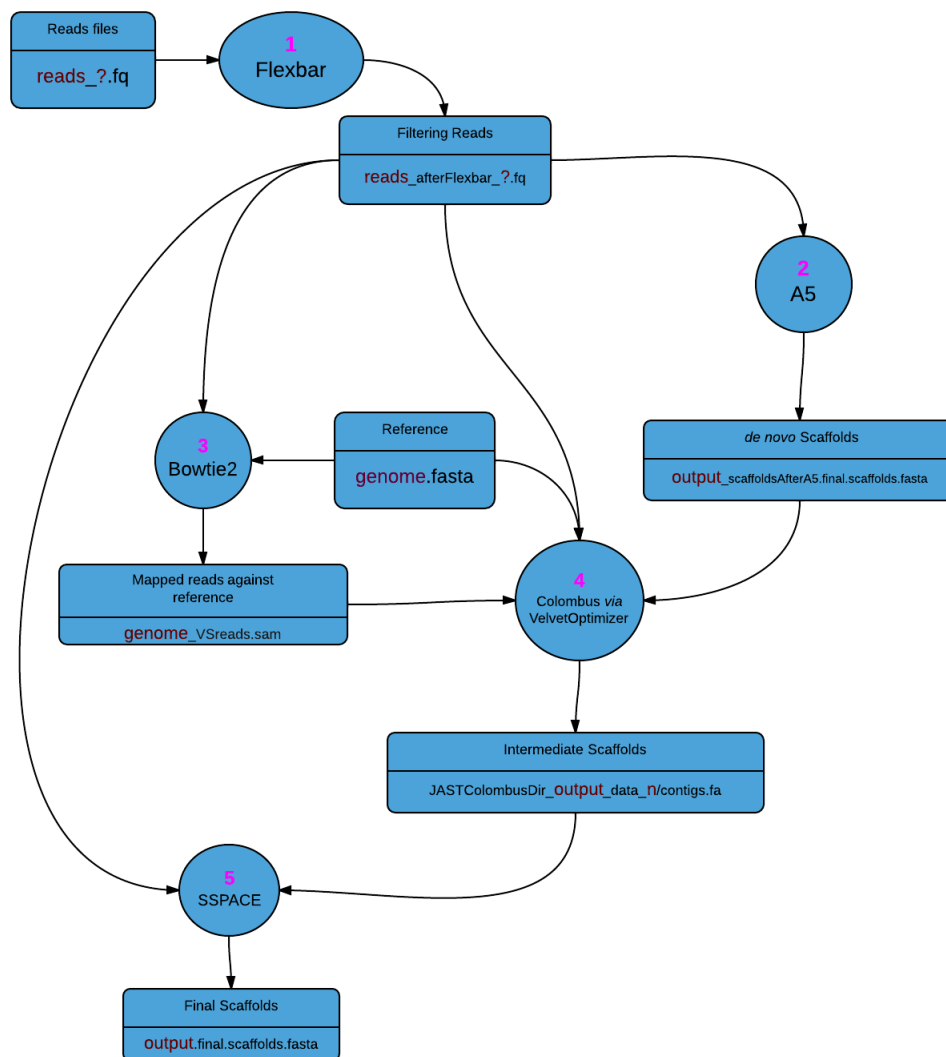


Figure 3: Complete pipeline of JAST.

3 For programmers

Dependencies

JAST use the JSAP library v2.1 downloaded in June 2014, under LGPL licence : <http://www.martiansoftware.com/jsap/> . You need to download it if you want to generate you own JAR file from source code or to modify source code.

Software architecture

In JAST, all commands inherit from the **Command** abstract class (see 4), has the following attributs & methods :

- Attributs :
 - command (String) : The executable command i.e. for flexbar it's "flexbar"
 - forbiddenOptions (String[]) : Options already used by JAST (users aren't allowed to use them).
 - totalCommand (List<String>) : All the command with executable and options.
 - config (Path) : The config file, can be null.
 - outputFile (Path) : The outputFile.
- Methods :
 - exec : Execute the command.
 - getOutPutFile : get the output file.
 - getCommand : get the executable command.

The source code can be found here : <https://github.com/AgResearch/JAST/tree/master/src> and the Javadoc can be found here : <https://github.com/AgResearch/JAST/tree/master/Javadoc> .

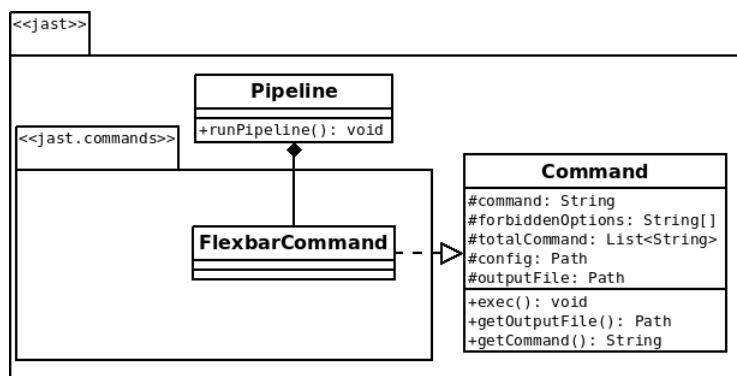


Figure 4: Class diagram example for one command (Flexbar).

Considering that bioinformatics software evolve rapidly in the field of sequencing, it could be usefull to add a new software to the pipeline. Here is how do so :

```
1 package jast.commands;
  import java.nio.file.Path;
3  import java.nio.file.Paths;
  import jast.Command;
5  public class newCommand extends Command {
      public newCommand(Path config,String [] arrayOfForbbidenOptions) {
7          super(config,arrayOfForbbidenOptions);
          command="foo";
9          // no need to add command to totalCommand
          totalCommand.add("-bar");
11         totalCommand.add("myArgumentForBarOption");
      }
13 }
```