# Tutorial

# PIPEMB-WDL

Index

[1. Preparing the input sequence files: 1](#_Toc80805252)

[2. Preparing the JSON file. 2](#_Toc80805253)

[3. Execution: 2](#_Toc80805254)

[4. Output directory 2](#_Toc80805255)

[5. File system structure used for examples 3](#_Toc80805256)

[5.1. Execution of germline short variant call, with preprocessing and annotation. 4](#_Toc80805257)

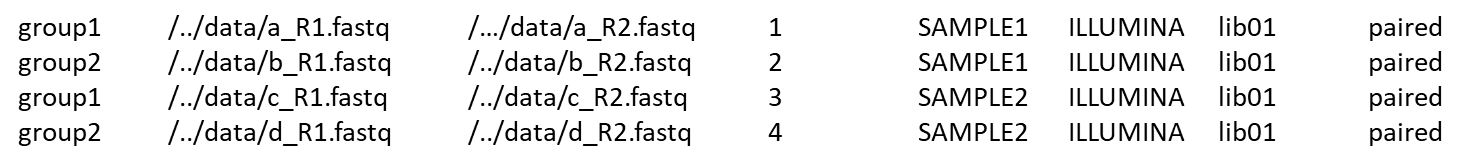
## Preparing the input sequence files:

The input file is formatted as TSV, which is a **simple text format for storing data in** a tabular structure. It is formed by the following columns:

1. Read group identifier
2. Sequence file address ( .ubam, .fastq extensions)
3. Sequence file address 2 ( for pair-end .fastq files)
4. Line identifier (unique for each line)
5. Sample name, same for all lines of the same sample
6. Sequencing platform
7. Library preparation
8. Pairing: "paired"
9. Sample type: "tumor" or "normal"
10. Paired sample: if you are going to do a tumor/normal somatic study, in this column put the name of the paired sample

Each line represents a reads group.

An example of a multi-sample and multi-read group is given in the next Figure.



## Preparing the JSON file.

* Common options and name convention for template configurations:

The JSON files determines what will be executed in the workflow. To recognize what steps are set, a convention for the name of file is followed. For example, we have: pr1g1\_SM\_Fl\_jn0s0pn0f1v1.json, what that means? The nomenclature used is:

* 0: disabled , 1: enabled
* pr: pre-processing
* g: germline
* SM: single sample mode
* MM: multisample mode
* Fl: filtering
* jn: joint genotyping
* s: somatic
* pn: create panel of normal
* f: filter with Funcotator
* v: filter with VEP
* ++/pp: filter with Funcotator with optional MAF and VCF for independent samples

## Execution:

* + Slurm script (command execution)
  + Execution time, output log (when you know that everything was successful)

## Output directory

The structure of the results in the selected directory (- output\_dir parameter) is the following:

* ***bam***: contains bam files resulting from preprocessing. Optional output, if it is used (do\_preprocessing = true, default).
* ***germline/somatic\_vcfs***: contains the vcf files resulting from germline/somatic variants call (HaplotypeCaller + CNNScoreVariants + FIlterVariantTranches / Program Mutect2 + FilterMutectCalls). Note: In the case of somatic, the filename has a T in front of it, but the truth is a vcf with the result for normal and tumor sample from the same sample. Optional output, if it is used (do\_<germline/somatic>\_short\_variant\_discovering = true) .
* ***germline/somatic\_PASS***: filtered vcf, contains only those variants that have "PASS" in the filter column. Optional output, if variant call or annotation is set.
* ***germline/somatic\_vcfs\_merged***: Contains the file final\_vcf\_all\_samples.vcf. It is a multisample vcf, containg the variants present in the germline/somatic\_PASS folder in a single file. Optional output, if variant call or annotation is set, and it is a multisample study. -\*\* *germline/somatic\_funcotator\_annot*\*\*: Contains the file final\_vcf\_all\_samples annotated by Funcotator. Optional output, if it is used (germline/somatic\_annot\_with\_funcotator = true)
* ***germline/somatic\_funcotator\_indep\_samples\_annot***: Contains a one file for each sample annotated by Funcotator using VCF format. Optional output, if it is used (germline/somatic\_annot\_with\_funcotator\_add\_allsamples = true)
* ***germline/somatic\_funcotator\_maf\_annot***: Contains a one file for each sample annotated by Funcotator using MAF format. Optional output, if it is used (germline/somatic\_annot\_with\_funcotator\_add\_maf = true)
* ***germline/somatic\_vep\_annot***: Contains file final\_vcf\_all\_samples annotated by Funcotator (if defined) and by VEP. Final file resulting from the workflow. Optional output, if it is used (germline/somatic\_annot\_with\_vep = true)

## File system structure used for examples

To illustrate an example of execution, was created an file structured composed by four directories located in /data04/tools/PIPEMB/homologacao/PIPEMB-WDL/TUTORIAL

* *configs* contains generic JSON files with example of main common options for workflows.
* *data:* contains the data used in the tutorial
* *execution:* directory from which the workflow must be executed for this tutorial. Contains the output log, generated for each execution and the Cromwell directory used during the execution.
* *result:* directory used to copy final outputs.

## Execution of germline short variant call, with preprocessing and annotation.

**Data:**

2 samples, NA12878\_20k, WGS Fastq format, from GATK test data.

Data source link: <https://console.cloud.google.com/storage/browser/gatk-test-data/wgs_fastq/NA12878_20k;tab=objects?organizationId=548622027621&project=broad-dsde-outreach&prefix=&forceOnObjectsSortingFiltering=false>

Script used to download the data: /data04/tools/PIPEMB/homologacao/PIPEMB-WDL/TUTORIAL/data/gatk-test-data/wgs-fastq/NA12878\_20k/getting\_data\_tutorial.sh

The two samples will be used to execute the short variant call. One sample is composed by two read group and other by one. Each read group is par-end data. The sequence input file used is: /data04/tools/PIPEMB/homologacao/PIPEMB-WDL/TUTORIAL/data/gatk-test-data/wgs-fastq/NA12878\_20k/ NA12878\_20k\_input\_data.tsv

**Configuration:**

The configuration file used is:

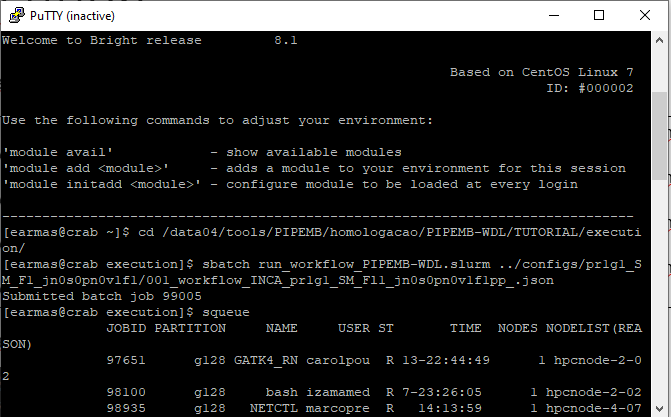
/data04/tools/PIPEMB/homologacao/PIPEMB-WDL/TUTORIAL/configs/pr1g1\_SM\_F1\_jn0s0pn0v1f1/ 001\_workflow\_INCA\_pr1g1\_SM\_Fl1\_jn0s0pn0v1f1pp\_.json

**Execution:**

1. Situate in execution directory

cd /data04/tools/PIPEMB/homologacao/PIPEMB-WDL/TUTORIAL/execution

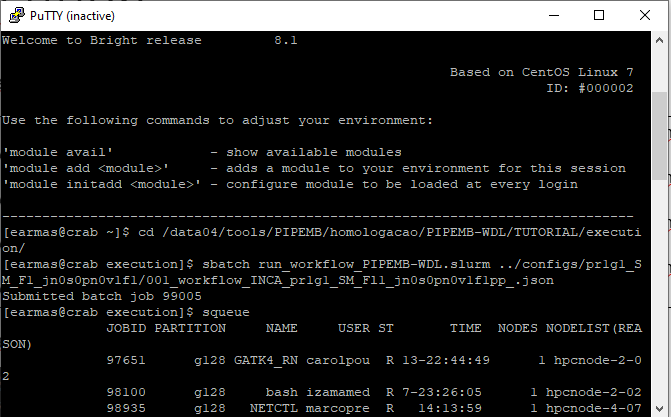
1. Call Slurm script passing the JSON file as parameter



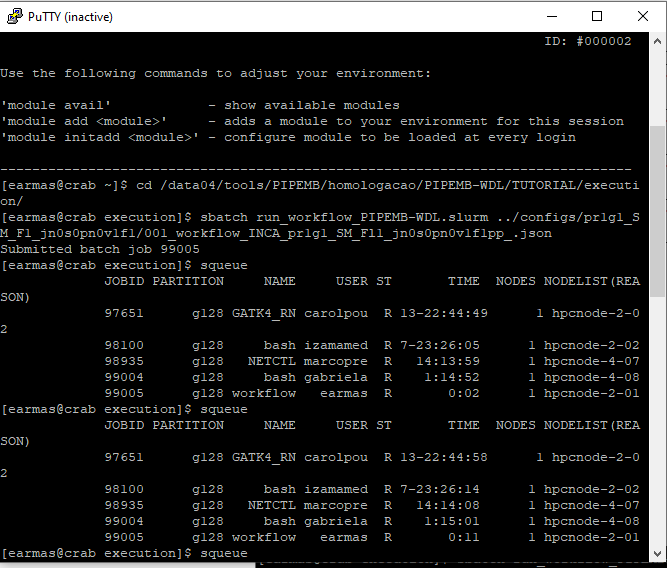
sbatch run\_workflow\_PIPEMB-WDL.slurm ../configs/pr1g1\_SM\_F1\_jn0s0pn0v1f1/ 001\_workflow\_INCA\_pr1g1\_SM\_Fl1\_jn0s0pn0v1f1pp\_.json

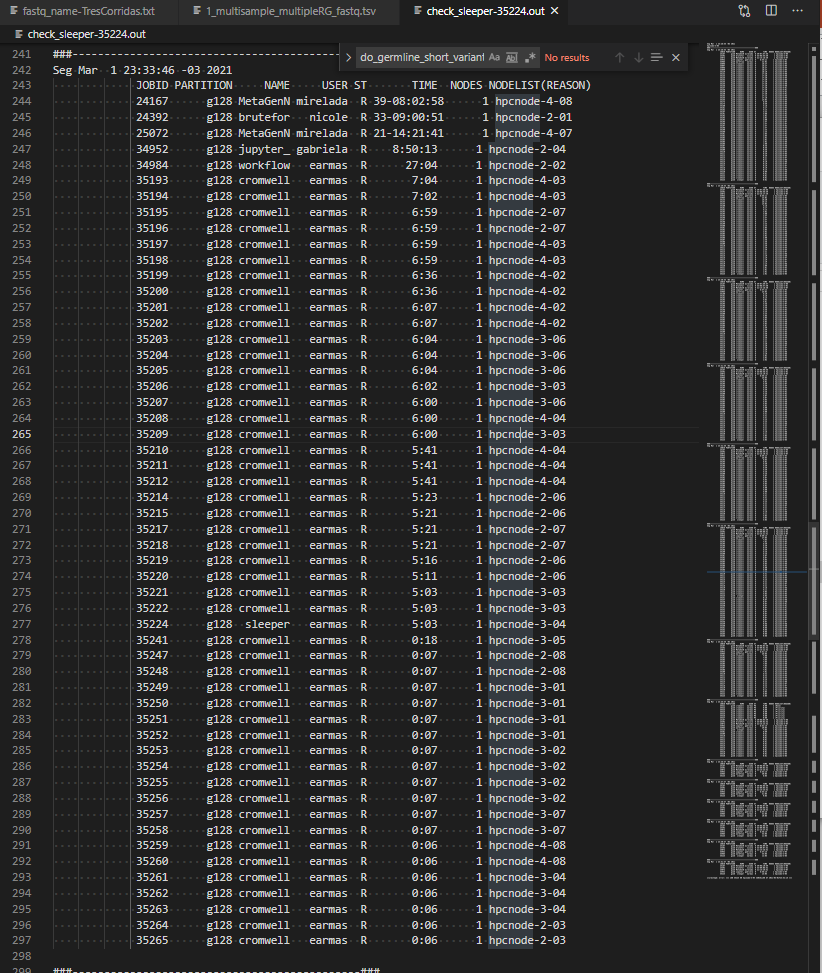
1. Following the workflow execution:

After starting the execution, the console shows the id number of job corresponding to the current execution. The Cromwell log is save at file named *workflow-<job id number>.log.* This log allows to know the current state of execution and when the workflow finished successfully. An example is illustrated in the following Figure.



In addition, it is possible to see the status of the Slurm queue (squeue command), that is, the currently running jobs and where they are running. Cromwell launches one job per task, and also the main workflow job. An example is shown in the following Figure:





1. View final outputs:

The configured output directory is /data04/tools/PIPEMB/homologacao/PIPEMB-WDL/TUTORIAL/results/001. The final outputs will appear in this directory, containing the corresponding subdirectories and files for the executed workflow phases.

The following videos show previous steps:

* + [Part 1](tutorial/videos/Part1_INCA.mp4). Presenting the structure of directories and JSON files.
  + [Part 2](tutorial/videos/Part2_INCA.mp4). Executing
  + [Part 3](tutorial/videos/Part3_INCA.mp4). Executing process. Output directory
  + [Part 4](tutorial/videos/Part4_INCA.mp4). Output directory