

Tutorial

PIPEMB-WDL

Index

1. Preparing the input sequence files:	1
2. Preparing the JSON file.	2
3. Execution:	2
4. Output directory	2
5. File system structure used for examples	3
5.1. Execution of germline short variant call, with preprocessing and annotation.	4

1. 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.

group1	../data/a_R1.fastq	../data/a_R2.fastq	1	SAMPLE1	ILLUMINA	lib01	paired
group2	../data/b_R1.fastq	../data/b_R2.fastq	2	SAMPLE1	ILLUMINA	lib01	paired
group1	../data/c_R1.fastq	../data/c_R2.fastq	3	SAMPLE2	ILLUMINA	lib01	paired
group2	../data/d_R1.fastq	../data/d_R2.fastq	4	SAMPLE2	ILLUMINA	lib01	paired

2. 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

3. Execution:

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

4. 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, containing 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)

5. 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.

5.1. 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_F11_jn0s0pn0v1f1/001_workflow_INCA_pr1g1_SM_F11_jn0s0pn0v1f1pp_.json

Execution:

- Situate in execution directory

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

- Call Slurm script passing the JSON file as parameter

```
[earmas@crab execution]$ sbatch run_workflow_PIPEMB-WDL.slurm ../configs/pr1g1_SM_F11_jn0s0pn0v1f1/001_workflow_INCA_pr1g1_SM_F11_jn0s0pn0v1f1pp_.json
Submitted batch job 99005
```

```
sbatch run_workflow_PIPEMB-WDL.slurm ../configs/prlg1_SM_F1_jn0s0pn0v1f1/001_workflow_INCA_prlg1_SM_F11_jn0s0pn0v1f1pp_.json
```

iii. 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- \langle job id number \rangle .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.

```
[earmas@crab execution]$ sbatch run_workflow_PIPEMB-WDL.slurm ../configs/prlg1_SM_F1_jn0s0pn0v1f1/001_workflow_INCA_prlg1_SM_F11_jn0s0pn0v1f1pp_.json
Submitted batch job 99005
```

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:

```
[earmas@crab ~]$ cd /data04/tools/PIPEMB/homologacao/PIPEMB-WDL/TUTORIAL/execution/
[earmas@crab execution]$ sbatch run_workflow_PIPEMB-WDL.slurm ../configs/prlg1_SM_F1_jn0s0pn0v1f1/001_workflow_INCA_prlg1_SM_F11_jn0s0pn0v1f1pp_.json
Submitted batch job 99005
[earmas@crab execution]$ squeue
```

	JOBID	PARTITION	NAME	USER	ST	TIME	NODES	NODELIST (REASON)
2	97651	g128	GATK4_RN	carolpou	R	13-22:44:49	1	hpcnode-2-0
	98100	g128	bash	izamamed	R	7-23:26:05	1	hpcnode-2-02
	98935	g128	NETCTL	marcopre	R	14:13:59	1	hpcnode-4-07
	99004	g128	bash	gabriela	R	1:14:52	1	hpcnode-4-08
	99005	g128	workflow	earmas	R	0:02	1	hpcnode-2-01

JOBID	PARTITION	NAME	USER	ST	TIME	NODES	NODELIST(REASON)
24167	g128	MetaGenN	mirelada	R	39-08:02:58	1	hpcnode-4-08
24392	g128	brutefor	nicole	R	33-09:00:51	1	hpcnode-2-01
25072	g128	MetaGenN	mirelada	R	21-14:21:41	1	hpcnode-4-07
34952	g128	jupyter_	gabriela	R	8:50:13	1	hpcnode-2-04
34984	g128	workflow	earmas	R	27:04	1	hpcnode-2-02
35193	g128	cromwell	earmas	R	7:04	1	hpcnode-4-03
35194	g128	cromwell	earmas	R	7:02	1	hpcnode-4-03
35195	g128	cromwell	earmas	R	6:59	1	hpcnode-2-07
35196	g128	cromwell	earmas	R	6:59	1	hpcnode-2-07
35197	g128	cromwell	earmas	R	6:59	1	hpcnode-4-03
35198	g128	cromwell	earmas	R	6:59	1	hpcnode-4-03
35199	g128	cromwell	earmas	R	6:36	1	hpcnode-4-02
35200	g128	cromwell	earmas	R	6:36	1	hpcnode-4-02
35201	g128	cromwell	earmas	R	6:07	1	hpcnode-4-02
35202	g128	cromwell	earmas	R	6:07	1	hpcnode-4-02
35203	g128	cromwell	earmas	R	6:04	1	hpcnode-3-06
35204	g128	cromwell	earmas	R	6:04	1	hpcnode-3-06
35205	g128	cromwell	earmas	R	6:04	1	hpcnode-3-06
35206	g128	cromwell	earmas	R	6:02	1	hpcnode-3-03
35207	g128	cromwell	earmas	R	6:00	1	hpcnode-3-06
35208	g128	cromwell	earmas	R	6:00	1	hpcnode-4-04
35209	g128	cromwell	earmas	R	6:00	1	hpcnode-3-03
35210	g128	cromwell	earmas	R	5:41	1	hpcnode-4-04
35211	g128	cromwell	earmas	R	5:41	1	hpcnode-4-04
35212	g128	cromwell	earmas	R	5:41	1	hpcnode-4-04
35214	g128	cromwell	earmas	R	5:23	1	hpcnode-2-06
35215	g128	cromwell	earmas	R	5:21	1	hpcnode-2-06
35217	g128	cromwell	earmas	R	5:21	1	hpcnode-2-07
35218	g128	cromwell	earmas	R	5:21	1	hpcnode-2-07
35219	g128	cromwell	earmas	R	5:16	1	hpcnode-2-06
35220	g128	cromwell	earmas	R	5:11	1	hpcnode-2-06
35221	g128	cromwell	earmas	R	5:03	1	hpcnode-3-03
35222	g128	cromwell	earmas	R	5:03	1	hpcnode-3-03
35224	g128	sleepers	earmas	R	5:03	1	hpcnode-3-04
35241	g128	cromwell	earmas	R	0:18	1	hpcnode-3-05
35247	g128	cromwell	earmas	R	0:07	1	hpcnode-2-08
35248	g128	cromwell	earmas	R	0:07	1	hpcnode-2-08
35249	g128	cromwell	earmas	R	0:07	1	hpcnode-3-01
35250	g128	cromwell	earmas	R	0:07	1	hpcnode-3-01
35251	g128	cromwell	earmas	R	0:07	1	hpcnode-3-01
35252	g128	cromwell	earmas	R	0:07	1	hpcnode-3-01
35253	g128	cromwell	earmas	R	0:07	1	hpcnode-3-02
35254	g128	cromwell	earmas	R	0:07	1	hpcnode-3-02
35255	g128	cromwell	earmas	R	0:07	1	hpcnode-3-02
35256	g128	cromwell	earmas	R	0:07	1	hpcnode-3-02
35257	g128	cromwell	earmas	R	0:07	1	hpcnode-3-07
35258	g128	cromwell	earmas	R	0:07	1	hpcnode-3-07
35259	g128	cromwell	earmas	R	0:06	1	hpcnode-4-08
35260	g128	cromwell	earmas	R	0:06	1	hpcnode-4-08
35261	g128	cromwell	earmas	R	0:06	1	hpcnode-3-04
35262	g128	cromwell	earmas	R	0:06	1	hpcnode-3-04
35263	g128	cromwell	earmas	R	0:06	1	hpcnode-3-04
35264	g128	cromwell	earmas	R	0:06	1	hpcnode-2-03
35265	g128	cromwell	earmas	R	0:06	1	hpcnode-2-03

iv. 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](#). Presenting the structure of directories and JSON files.
- [Part 2](#). Executing
- [Part 3](#). Executing process. Output directory
- [Part 4](#). Output directory