

May 11, 2022

Bioinformatic workflow for NGS data control

DOI

<https://dx.doi.org/10.17504/protocols.io.8epv59bnjg1b/v1>

Bioinformatic
workflow for
NGS data
control

Khalid El Moussaoui¹

¹AIDS Reference Laboratory, Department of Clinical Microbiology, University Hospital of Liege, 4000 Liege, Belgium



Khalid El Moussaoui

AIDS Reference Laboratory, Department of Clinical Microbiolo...

Create & collaborate more with a free account

Edit and publish protocols, collaborate in communities, share insights through comments, and track progress with run records.

Create free account

OPEN  ACCESS



DOI: <https://dx.doi.org/10.17504/protocols.io.8epv59bnjg1b/v1>

Protocol Citation: Khalid El Moussaoui 2022. Bioinformatic workflow for NGS data control. **protocols.io**
<https://dx.doi.org/10.17504/protocols.io.8epv59bnjg1b/v1>

License: This is an open access protocol distributed under the terms of the **[Creative Commons Attribution License](#)**, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited



Protocol status: In development

We are still developing and optimizing this protocol

Created: May 11, 2022

Last Modified: May 11, 2022

Protocol Integer ID: 62429

Keywords: ngs data control workflow, bioinformatic workflow, illumina novaseq6000, workflow, sequencing, data integrity, work environment configuration, ng, architected apple silicon processor

Disclaimer

DISCLAIMER – FOR INFORMATIONAL PURPOSES ONLY; USE AT YOUR OWN RISK

The protocol content here is for informational purposes only and does not constitute legal, medical, clinical, or safety advice, or otherwise; content added to protocols.io is not peer reviewed and may not have undergone a formal approval of any kind. Information presented in this protocol should not substitute for independent professional judgment, advice, diagnosis, or treatment. Any action you take or refrain from taking using or relying upon the information presented here is strictly at your own risk. You agree that neither the Company nor any of the authors, contributors, administrators, or anyone else associated with protocols.io, can be held responsible for your use of the information contained in or linked to this protocol or any of our Sites/Apps and Services.

Abstract

Workflow for data integrity and quality control of high throughput sequencing on Illumina NovaSeq6000. The analyses are performed on macOS Monterey 12.3.1 running on an ARM-architected Apple Silicon processor. This workflow considers that the user directory (~/) is structured as seen in the "work environment configuration" protocol. To avoid error messages, please follow this protocol and set up your computer before starting.



Activation of the environment

- 1 Open a terminal window.

Software

Terminal

NAME

macOS Monterey 12.3.1

OS

Apple Inc.

DEVELOPER

- 2 Activate the previously created QC_env environment by typing the following command in the terminal :

Command

new command name`conda activate QC_env`

Data integrity check

- 3 Considering that the .gz archive downloaded from the GIGA servers has been unzipped under ~/fastq_files, that the original_md5.txt file has been stored under ~/md5 and that the python & R scripts previously created are stored under ~/KE_utilities, type the following command in the terminal to recompute the md5 hash and store it in a new file under ~/md5

**Command****new command name**

```
md5 ~/fastq_files/* > ~/md5/recomputed_md5.txt
```

- 4 After generating the ~/md5/recomputed_md5.txt file, type the following command in the terminal to launch the python script that allows the data integrity check :

Command**new command name**

```
python3 ~/KE_utilities/data_integrity_checker.py
```

This script can be downloaded on GitHub : https://github.com/elmoussaoui-k/drylab_workflow/blob/b41c0a03d7a3fdb023a94b331cab5460d706b6c8/data_integrity_checker.py

- 5 Specify the path to the original_md5.txt file and then to the recomputed_md5.txt file :

**Command**

new command name

```
***** DATA INTEGRITY CHECKER *****
```

```
Please enter the path to original_md5.txt :
```

```
/users/khalid/md5/original_md5.txt
```

```
Please enter the path to recomputed_md5.txt :
```

```
/users/khalid/md5/recomputed_md5.txt
```

```
-----
```

Run fastQC

- 6 Start the fastQC analysis on all existing files in the ~/fastq_files directory in recursive mode using "*". Moreover, the addition of the --outdir option allows to specify an output directory for the reports generated by fastQC. This generates an individual .html report for each file.

Command

fastqc version : v. 0.11.9

new command name

```
fastqc ~/fastq_files/* --outdir ~/fastqc_reports/
```

- 7 The generated reports can be opened by typing the following command in the terminal :

**Command**

new command name

```
open ~/fastqc_reports/KE0xx_R1_fastqc.html
```

Run multiQC

- 8 To summarize the reports generated with fastQC into a single report, run multiQC. To do this, type the following command in the terminal :

Command

multiqc version : v. 1.12

new command name

```
multiqc ~/fastqc_reports --outdir ~/multiqc_report
```

- 9 The generated report can be opened by typing the following command in the terminal :

Command

new command name

```
open ~/multiqc_report/multiqc_report.html
```



Filter reads with fastp

- 10 The reads can be filtered automatically with fastp. Just launch the program, specify the 2 .fastq.gz files (R1 and R2) as input and specify the name and location of the 2 processed files. Adding the -h option allows to specify a folder for the HTML report. The option -j " " allows to cancel the creation of the JSON report. The -R option allows to give a name to the generated HTML report.

Command

fastp version : v. 0.23.2

new command name

```
fastp -i ~/fastq_files/KE0xx_R1.fastq.gz
-I ~/fastq_files/KE0xx_R2.fastq.gz
-o ~/fastp/cleaned_fastq_files/KE0xx_R1_clean.fastq.gz
-O ~/fastp/cleaned_fastq_files/KE0xx_R2_clean.fastq.gz
-h ~/fastp/fastp_reports/KE0xx_fastp_report.html
-j ""
-R "Fastp report : KE0xx"
```