Pre-Alignment QC Workflow

User Documentation and Tutorial

CCRIFX Bioinformatics Core

Date: August 29th, 2013

Purpose

This workflow is used to merge the raw Fastq files generated from sequencing runs based on the Illumina Fastq naming scheme, followed by automated job submission for running FastQC [1] on the merged FastQ files. In addition, Fastq files can be trimmed using trimmomatic [2] if necessary.

Introduction

Raw FastQ files are generated from sequencing runs at the Sequencing facility (SF). The sequencing facility demultiplexes these using Casava 1.8. The investigator or the CCRIFX core is notified about the completion of the sequencing run and the report provided by SF contains the location of the FastQ files.

Illumina FastQ files use the following naming scheme:

<sample name>_<barcode sequence>_L<lane (0-padded to 3 digits)>_R<read number>_<set number (0-padded to 3 digits>.fastq.gz

For example, the following is a valid FastQ file name:

NA10831_ATCACG_L002_R1_001.fastq.gz

In the case of non-multiplexed runs, <sample name> will be replaced with the lane numbers (lane1, lane2, ..., lane8) and <barcode sequence> will be replaced with "NoIndex".

For further details on the Illumina naming convention, please refer to [4] under References below.

Once the QC.07.09.2013.pl completes execution, the resulting merged files are of the form:

<sample name>_<barcode sequence>_L<lane (0-padded to 3 digits)>_R<read number>_all.fastq

Users need to inspect the fastqc results and determine how to trim the raw reads if trimming is necessary. The QC.07.19.2013.pl will verify the pairing if the data are paired fastq and do the trimming, followed by fastqc quality check.

Prerequisites

a) This documentation assumes that the user has either a Biowulf or Moab account
b) Successfully source ccrifx_sys_setup.sh
c) The tutorial provided below assumes that execution is carried out on Biowulf

Input

Using QC.07.09.2013.pl to aggregate chunks, validate pairing and generate FastQC reports of the raw fastq data

a) A full-path to the directory containing the FastQ "chunks"
b) A full-path to the output directory where the resulting merged FastQ files and FastQC output will appear. The directory should not already exist.
c) Specification of the queue mode [y or n]. If set to yes, the execution of merging the FastQ files is submitted as a batch job on the cluster.

In case the trimming is needed using QC.07.19.2013.pl

a) A full-path to the directory containing the FastQ files that have been generated using QC.07.09.2013.pl script.
b) A full-path to the output directory where the trimmed FastQ files and FastQC output will appear. The directory should not already exist.
c) Specification of the queue mode [y or n]. If set to yes, the execution of merging the FastQ files is submitted as a batch job on the cluster.
d) The quality encoding either phred 33 or 64
e) Trimmomatics option
f) Whether the log file of trimming is needed.

Output

Output of QC.07.09.2013.pl

a) Merged FastQ files for each sample and read end. Once the workflow completes execution, the resulting merged files are of the form: *<sample name>_<barcode sequence>_L<lane (0-padded to 3 digits)>_R<read number>_all.fastq*

b) FastQC output for each sample and read-end. FastQC output is grouped into a directory and contains the zipped output file and the job files created by the workflow. The user can unzip the zipped FastQC file in order to visualize the FastQCreport in html format.

Output of QC.07.19.2013.pl if used:

a) Trimmed fastq files for each sample under trimming sub-directory: sample-name.fastq.trim.fastq

b) FastQC output for each trimmed fastq file.

Tutorial

In the tutorial the unix commands are in **bold font**.

1. The test data sets are located under /data/CCRIFX/QC_WF_test/pair. There are paired-end fastq chunks from two data sets.

2. Add the environment to your path by using the following command

$source /data/CCRIFX/RELEASE_08.29.2013/ccrifx_root_committed/ccrifx_sys_setup.sh

After typing the above command, the $PATH environment variable is updated. You can check for this using the following unix commands (**echo $PATH**), an environment variable $CCRIFX_SYS_ROOT should be defined. You can check the environment variable $CCRIFX_SYS_ROOT is set correctly by using the following unix command (**ls $CCRIFX_SYS_ROOT).** The benefit of setting these environments is so that you have access to the scripts plus dependencies without having to refer to them via full path. If any error message is returned whilst setting environments – please report the error using the issue-reporting template.

3. The script QC.07.09.2013.pl expects the names of two directories. An input directory that holds the chunks and a user specified output directory for the merged fasta files and the FastQC reports. If the script is called with none or incorrect parameters – the program will output information with correct command line usage for the program.

**$ QC.07.09.2013.pl **

**--fqdir /data/CCRIFX/QC_WF_test/pair/ **

**--outdir $PWD/QC070913_pair **

--queue y

4. Check your jobs. Use the following command to see jobs submitted by the script and queued for execution, please read [3] for more information.

qstat –u <username>

5. What to expect after job submission using QC.07.09.2013.pl?

There are several unix commands to help track whether everything is going to plan. Use the **qstat** command to check for job status on the biowulf processors

and find out how many of your jobs are running on the processors and how much processing-time each job has been used.

Check the log files in the output directory. Review the log.txt file and the *.job file and subdirectories in the output directory. In this case the output directory is /data/CCRIFX/QC_WF_test/QC070913_pair. Review the output files e.g., *_all.fastq, FastQC files and the FastQC zip files. Review the merged FastQ files and any other filetype that appears. In the subdirectory for each sample there should be two FastQ files one for each sample (read1 and read2). The validityReport.txt file tells user whether the pairing is validated or if the input fastq files are just single-ended.

You will see the files and what time data was last added to each file. At some point four pairs of FastQ files should merge successfully. The pairs should have the same number of bytes. After this step has completed, the script will progress to submitting FastQC jobs for each of the completed merged FastQ files. One FastQC report will be created for read1 and read2. The FastQC jobs will take about 30 minutes to run to completion. The time depends on the size of the merged FastQ files and how busy the nodes are. For each sample you will have a zip file and another subdirectory containing your FastQC report.

If the trimming is needed, users can proceed to the QC.07.19.2013.pl script.

1.Aggregated fastq data are located under /data/CCRIFX/QC_WF_test/QC070913_pair/QC070913_pair. These data are paired-end fastq generated from QC.07.09.2013.pl script.

2. Add the environment to your path and check the installation of the tools needed by using the following command in case this has not been done yet.

$source /data/CCRIFX/RELEASE_08.29.2013/ccrifx_root_committed/ccrifx_sys_setup.sh

3. Command line example, the –-outdir should not be existing:

**QC.07.19.2013.pl **

**--phred 33 **

**--trimlog y **

**--trimomaticopts 'HEADCROP:1 LEADING:2' **

**--fqdir /data/CCRIFX/QC_WF_test/QC070913_pair/QC070913_pair **

**--outdir $PWD/QC071913_pair **

--queue y

Users need to provide the options for the trimmomatic program wrapped in this script.

4. Check your jobs. Use the following command to see jobs submitted by the script and queued for execution

qstat –u <username>

5. What to expect after job submission?

Check the log files in the output directory. Review the log.txt file and the *.job file and subdirectories in the output directory. To see whether the fastq files are properly paired data, check the validityReport.txt. The trimming will take a while, typically ~ 5 hours for paired-end data. The trimming log file may be 300 Mb up to 1 Gb in size (gz format).

References

1. **FastQC documentation**
   http://www.bioinformatics.babraham.ac.uk/projects/fastqc/

2. **Trimmomatic manual and reference**
   http://www.usadellab.org/cms/?page=trimmomatic

   Lohse M, Bolger AM, Nagel A, Fernie AR, Lunn JE, Stitt M, Usadel B. (2012) RobiNA: a user-friendly, integrated software solution for RNA-Seq-based transcriptomics. Nucleic Acids Res. 2012 Jul;40(Web Server issue):W622-7.

3. **Biowulf documentation for monitoring jobs on the queue**
   http://biowulf.nih.gov/user_guide.html#monitor

4. **Illumina naming scheme and FastQ file contents**
   http://support.illumina.com/help/SequencingAnalysisWorkflow/Content/Vault/Informatics/Sequencing_Analysis/CASAVA/swSEQ_mCA_FASTQFiles.htm

5. **Unix cheat sheets**
   http://www.cyberciti.biz/tips/linux-unix-commands-cheat-sheets.html

6. **FASTQC on biowulf:**
   http://biowulf.nih.gov/apps/fastqc.html

7. **trimmomatic on biowulf:**
   http://biowulf.nih.gov/apps/trimmomatic.html

FAQ

1) **Can this workflow be used on FastQ files from any sequencing application (eg. ChIP-seq, RNA-seq, Exome-seq etc.)** Yes, this is a general QC workflow that is applicable to FastQ files generated for any sequencing application. The only requirement is that the FastQ files follow the Illumina naming convention as described in the Introduction section and can be processed by FastQC.

2) **Can this workflow be used on a FastQ files that are already merged using some other application?** Yes, this workflow can be used on merged FastQ files. Again, the naming convention should be adhered to for the merged FastQ files.

3) **Which output file should I download to visualize the FastQC report?** The zipped file created by the workflow contains all the files required to

view the html version of the report. So it is best to download this zipped file locally in order to open the report in a browser.

4) **FastQC can be run on BAM files. Can this workflow be run on BAM files?** This version of the workflow currently supports only FastQ files.

Contact

If you identify areas where you would like to expand the user documentation or would like to make an update to the user documentation or provide comments/feedback please contact Yvonne Edwards or Parthav Jailwala.

CCRIFX Bioinformatics Core
Advanced Biomedical Computing Center (ABCC)
Information Systems Program
Leidos Biomedical Research, Inc.
Frederick National Laboratory for Cancer Research (FNLCR)
P. O. Box B, Frederick, MD 21702
Phone: 301.594.1395
Fax: 301.480.0391
http://ccrifx.cancer.gov

Citation

If you want to cite this workflow, please use the following link:

Pre-alignment QC workflow:
http://ccrifx.cancer.gov/apps/site/workflows_for_bioinformatics_analysis

This document was lasted updated Dec 13, 2013.