# Abstract

For most applications, short reads generated by next-generation sequencing technologies require pre-processing. This SOP describes a basic pre-processing workflow for data generated by CRC projects – including identification of potential issues with FastQC, cleaning of reads using trimmomatic and optional error correction for downstream assembly applications.

# Introduction

Short reads generated by next-generation sequencers will typically include non-genomic/non-transcriptomic nucleotides added by the sequencing library preparation (barcodes, sequencing adapters). In addition, quality of sequenced reads may vary along the length of individual reads, sometimes falling below acceptable values for certain applications. This SOP outlines the relevant steps required to clean read files for downstream processing.

# Requirements

## Input data

Paired-end reads generated on the Illumina Platform (HiSeq 2500, HiSeq 3000, HiSeq 4000, NextSeq and MiSeq).

## Software

Trimmomatic (0.36)

Trowel (0.2.0.4)

FastQC (0.11.3)

# Procedure

## Quality statistics and identification of issues

FastQC analyses individual FASTQ files to identify a range of potential errors, including low quality bases, presence of known adapter sequences and overrepresented kmers (identifying e.g. unclipped barcodes or unknown adapters).

FastQC results are normally delivered by the sequencing provider together with the raw data. If not, the syntax for generating such a report is as follows:

fastqc –o fastqc\_analysis <input file>

where <input file> is either the left or right mate (or single-end) fastq file (preferably in compressed gzip format). The results are available as HTML file (readable by any web browser) as well as a series of log files, compressed into a zip archive. A detailed explanation of how to read the output is available on the FastQC website: http://www.bioinformatics.babraham.ac.uk/projects/fastqc/

## Read trimming

Based on the results from the FastQC analysis, it may be necessary to clean the data. We recommend Trimmomatic as it, unlike most other tools, provides a reference file with commonly used sequencing adapters. Trimmomatic can perform a range of correction steps, including clipping of adapters as well as of low quality bases. Note that it is NOT recommended to perform quality trimming if the data ist to be used for de-novo assembly (genome or transcriptome). However, if the data will be used for highly sensitive variant calling (or similar), it is recommended to perform stringent clipping of all sub-par bases.

A basic Trimmomatic command will look as follows:

java –jar trimmomatic-0.36.jar PE –threads 8 my\_reads\_R1.fastq.gz my\_reads\_R2.fastq.gz my\_reads\_R1\_paired.fastq.gz my\_reads\_R1\_unpaired.fastq.gz my\_reads\_R2\_paired.fastq.gz my\_reads\_R2\_unpaired.fastq.gz ILLUMINACLIP:<ADAPTER\_FILE>:2:30:10 LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15

Here, <ADAPTER\_FILE> points to the appropriate adapter sequence file included with Trimmomatic. The last part of the command, specifying a sliding window clipping of lower quality bases, is optional and should be ommitted when performing assembly tasks.

## Error correction

Some assembly programs will benefit from error-corrected reads. This procedure is different from trimming and aims to correct individual positions without clipping using Kmer analysis or similar. Note that your assembler of choice (like Spades or Allpaths-LG) likely already contains a built-in error correction module, so in those cases it is not recommended to perform an extra correction step beforehand.

Trowel has proven as the most performant and easy to install option for the purpose of error correcting reads for assembly. The process for running Trowel includes generating a list of all files tob e processed and the choise of an appropriate kmer size for analysis (13-15 are recommended for larger eukaryotic genomes).

trowel –f files.txt –k 14

# Implementation

This workflow is implemented as pipeline “ReadPreprocessing.bpipe”, included with the CRC git code repository.

# Related Documents

# Revision history