SEQprocess: An automated and extendable pipeline framework for NGS data processing in R package

Contents

1 Background

Next-Generation Sequencing (NGS) technology is now widely used for biomedical research with various applications including identification of sequence variants from DNA or RNA sequences, and quantitation of RNA expression or DNA copy numbers. Previously, several softwares for NGS data processing pipelines have been released, however, rapid progress of the NGS applications and data processing methods urgently require prompt update of the pipelines such as the NCI genome data commons (GDC) used for processing the TCGA data. Recent clinical applications of NGS technology such as cell-free DNA, small RNA, cancer panel, and exosomal RNA sequencing data also require appropriate data processing pipelines. With this concern, we developed SEQprocess, a flexible and extendable framework that can provide NGS processing pipelines covering the GDC pipeline as well as the new data for clinical applications.

2 Introduction

Next-Generation Sequencing (NGS) technology is now widely used for biomedical research fields, and is extensively being used in the clinic (Kwon, et al., 2012). Applications of NGS technology include the identification of sequence variants of DNA or RNAs, and the quantitation of RNA abundances or DNA copy numbers. However, processing and analysis of NGS data still remain painful, because data processing procedure requires complex multiple steps and softwares. NGS data are generally processed by multiple processing steps including quality control, trimming, removal PCR duplication, sequence alignment, variant calling, and annotation, and each of the step requires different legacy softwares. To handle these complex processing steps, several softwares have been released. For example, NGS-pipe (Singer, et al., 2018) and NEAT (Schorderet, 2016) provide automated pipelines for NGS data analysis. Another tool systemPipeR also provides a NGS analysis workflow in R program that can be customized according to the various NGS applications such as whole-exome (WES), whole-genome (WGS) and transcriptome (RNA-seq) sequencing data (Backman and Girke, 2016). However, these tools do not handle the recently updated GDC pipelines which have been used in the processing of the TCGA data. Moreover, recent progress of clinical applications of the NGS data generate new platform data such as, cell free DNAs, exosomes, cancer panel, and single cells. These applications require analysis with customized setting for data quality control and processing. However, there is no comprehensive tools that can handle the recent clinical advancement in NGS technologies.
With this concern, we developed a SEQprocess that can provide customizable NGS processing pipelines covering the GDC pipelines as well as the new data for clinical applications. SEQprocess is implemented in an R program to provide an automated and user-friendly interface. SEQprocess provides a flexible customization framework by modularizing the sequences of the NGS processing steps. SEQprocess also provides six pre-customized pipelines which are widely used as standards in NGS data processing that can be executed easily by non-experts such as biomedical scientists.

3 Getting Started

3.1 System requirements

SEQprocess is developed as an R package which requires various R/Bioconductor libraries. Since the processed data are usually further analyzed for statistical assessment or visualization under R environment, R-compatible implementation of SEQprocess will facilitate the subsequent analyses for biological interpretation under R environment. The current version of SEQprocess supports Linux operating system, due to compatibility of the required softwares. Parallel computation on multi-core machines is also supported by using 'parallel' R package.

3.2 Installation

Source codes for SEQprocess are available at : https://github.com/omicsCore/SEQprocess

```
source("https://bioconductor.org/biocLite.R")
biocLite("GenomeInfoDbData")
biocLite("DelayedArray")

install.packages("devtools")
library(devtools)

install_github("omicsCore/SEQprocess")
```

3.3 Loading package and documentation

```
library("SEQprocess") # Loads the package
library(help="SEQprocess") # Lists package info
```

3.4 Required library

Processing	library
Quality Check	parallel
Trimming	parallel
Alignment	parallel
Remove Duplicates	parallel
Realignment	parallel
Variant Calling	parallel
Annotation	parallel
Copy number	parallel, sequenza
RNA quantitation	parallel, GenomicRanges
Make Set	parallel, Biobase, GenomicRanges, SummarizedExperiment
Report	parallel, limma, data.table, fastqcr, pander, knitr, png, grid, gridExtra, ggplot2, reshape2

3.5 External legacy softwares required to runused in the pipelines

3.51 Direct install external tools

SEQprocess implements six processing pipelines for the DNAseq and RNAseq pipelines. In DNAseq DSEQ) have a GDC, Custom, cfDNA pipelines and RNAseq(RSEQ) have GDC, Tuxedo, Small/Exome RNA pipelines. For each of the implemented pipelines in SEQprocess, following external softwares are required to be installed in the system :

Software	Version	Description	Use
FASTQC	v0.11.5	Check for the quality of fastq file	DSEQ, RSEQ
Trim Galore	v0.4.2	Trims the adapter and low phred score sequences	DSEQ, Tuxedo

Software	Version	Description	Use
Cutadapt	v1.11	Removes the adapter sequences, primers, poly-A tails and other types of unwanted sequence from high-throughput sequencing reads	miRseq
BWA	v0.7.15	Align of the fastq file to reference genome. Methods of bwa-mem & bwa-samse are used	DSEQ, miRseq
SAMtools	v0.1.18	Manipulates alignments indexing, converting and statistic	GDC
STAR	v2.5.2b	Aligns short and long RNA-seq reads	GDC
Tophat2	v2.1.1	A fast splice junction mapper for RNA-Seq reads	Tuxedo
Bowtie2	v2.2.9	Sequence alignment	miRseq
Picard	v2.17.4	Remove PCR duplicates and ordering	DSEQ
GATK	v3.7	Variant discovery and genotyping	DSEQ
SomaticSniper	v1.0.5.0	Identify single nucleotide positions that are different between tumor and normal (or in theory, any two bam files)	GDC
Varscan2	v2.4.3	Variant detection	GDC
Mutect2	v1.1.4	Somatic SNP and indel caller	GDC
MuSE	v1 Orc	Mutation calling allelic composition of the tumor and normal tissue at each reference base	GDC
Variant Effect Predictor	v91	VEP determines the effect of your variants on genes, transcripts, and protein sequence, as well as regulatory regions	GDC
ANNOVAR		Annotation of information to functionally genetic variants	Custom, cfDNA
HT-Seq	v0.6.1	Given a file with aligned sequencing reads and a list of genomic features, a common task is to count how many reads map to each feature	GDC, miRseq
Cufflinks	v2.2.1	Assembles transcriptomes from RNA-Seq data and quantifies their expression	Tuxedo

3.52 Install external tools using Conda

External software can also be installed using Conda:Bioconda. Bioconda is a channel for the Conda package manager specializing in bioinformatics software. All programs except the CRAN-R package, sequenza and ANNOVAR, can be installed using bioconda. We recommend installing the program using bioconda on Linux as follows :

(1) Install Miniconda by choosing the appropriate Python version and platform from the URL below. (Linux recommended) https://conda.io/miniconda.html

(2) Continue installing using the following command line with the installed Miniconda bash script.

```
bash Miniconda2-latest-Linux-x86_64.sh (bash script-path)
```

(3) Connect the previously installed Conda with Bioconda, which manages the Bioinformatics tools.

```
conda config --add channels bioconda
```

(4) Use Conda and bioconda to install the required software.

```
conda install <fastqc, trim-galore, bwa...>
```

(5) See Conda and Bioconda for details.
https://conda.io/docs/index.html
https://bioconda.github.io/index.html

3.6 Pre-customized pipelines

Current version of SEQprocess provides six different pre-customized standard pipelines, including the pipelines for GDC processing and the newly adapted clinical applications for cell-free DNAs (cfDNA) and exosomal miRNAs (Fig. 1). These pipelines can be run by one-step command that can be executed easily by non-expert users. For WGS/WES, a GDC compatible pipeline of TrimGalore-BWA-Picard- VarScan2-VEP is implemented. We also implemented a popularly used standard Custom pipeline of TrimGalore-BWA-Picard-GATK–ANNOVAR. In addition, SEQprocess can estimate allele frequencies for each variant by calculating the sequence read depths of the mutated and wild-type sequences with a GATK function gatk.depthOFcoverage. For liquid biopsied cfDNA or targeted sequencing data such as cancer panel, an optimized pipeline excluding the step for removal duplicates is provided, because the sequence reads of the cfDNAs are usually have same sequences. For barcoded data (BarSeq), the step for removal duplicates is performed using the barcode information.
For RNA-Seq data, a GDC pipeline STAR-Samtools-HTSeq pipeline is implemented. A popularly used standard pipeline Tuxedo (i.e.,Tophat2-Cufflinks) is also implemented. For miR-Seq data from exosomes, cells, or tissues, a pipeline of Cutadapt-BWA/bowtie2-HTSeq is implemented with optimized parameters.

4 System configuration

Configuration of the installed softwares and data can be managed simply by editing the `data/config.R` file. SEQprocess also supports parallel computation on multi-core machines by using 'multicore' R package.

4.1 Path configuration

4.1.1 Softwares

If the required softwares are installed on a specific directory, you must set a 'program.dir' variable.

```
program.dir="/data/program"
```

Then, the executable file paths of softwares should be setted on this script file. If you don't know the executable file path, type 'which fastqc' on the shell or system('which fastqc') on your R session.

```
------------------------------ **QualityChecek** --------------------------------
# fastqc
fastqc.path=file.path(program.dir, "bin", "fastqc")
------------------------------ **Trimming** -------------------------------------
# trim
trim_galore.path=file.path(program.dir,"bin","trim_galore")
# cutadapt
cutadapt.path=file.path(program.dir, "bin/cutadapt")
------------------------------ **Alignment** ------------------------------------
```

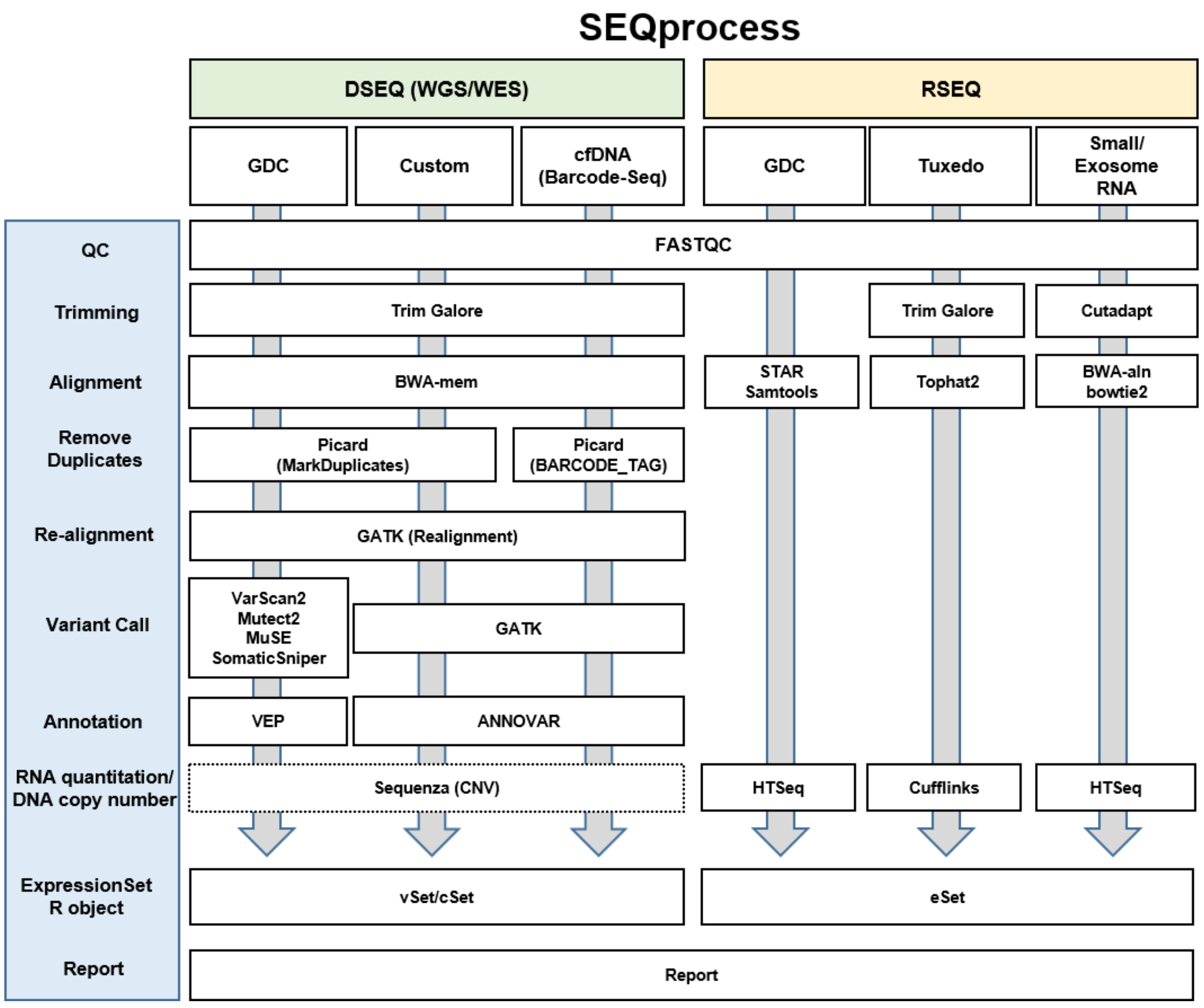
SEQprocess
DSEQ (WGS/WES)
RSEQ
GDC
Custom
cfDNA (Barcode-Seq)
GDC
Tuxedo
Small/ Exosome RNA
QC
FASTQC
Trimming
Trim Galore
Trim Galore
Cutadapt
Alignment
BWA-mem
STAR Samtools
Tophat2
BWA-aln bowtie2
Remove Duplicates
Picard (MarkDuplicates)
Picard (BARCODE_TAG)
Re-alignment
GATK (Realignment)
Variant Call
VarScan2 Mutect2 MuSE SomaticSniper
GATK
Annotation
VEP
ANNOVAR
RNA quantitation/ DNA copy number
Sequenza (CNV)
HTSeq
Cufflinks
HTSeq
ExpressionSet R object
vSet/cSet
eSet
Report
Report

Figure 1:

```
# bwa
bwa.path=file.path(program.dir, "bin", "bwa")
# samtools
samtools.path=file.path(program.dir, "bin", "samtools")
# STAR
STAR.path=file.path(program.dir, "bin/STAR")
# tophat2
tophat2=file.path(program.dir, "bin/tophat2")
# bowtie2
Sys.setenv("PATH"=paste(Sys.getenv("PATH"), "/data/program/bin/", sep=":"))
bowtie2.path=file.path(program.dir, "bin/bowtie2")
------------------------------ **RemoveDuplicate** ------------------------------
# picard
picard.path=file.path(program.dir, "picard/picard/build/libs/picard.jar")
------------------------------ **VariantCalling** -------------------------------
# GATK & MuTect2
GATK.path=file.path(program.dir, "gatk/3.7/GenomeAnalysisTK.jar")
# varscan2
Varscan.path=file.path(program.dir, "varscan/VarScan.v2.4.3.jar")
# somaticsniper
somaticsniper.path=file.path(program.dir, "bin/bam-somaticsniper")
# MuSE
MuSE.path=file.path(program.dir, "bin/MuSE")
------------------------------ **Annotation** -----------------------------------
# VEP
vep.path=file.path(program.dir, "bin/vep")
# ANNOVAR
vcf2annovar.pl=file.path(program.dir, "bin/convert2annovar.pl")
table_annovar.pl=file.path(program.dir, "bin/table_annovar.pl")
------------------------------ **Copy number** ------------------------
# Sequenza
sequenza.util=system.file("exec", "sequenza-utils.py", package="sequenza")
------------------------------ **RNAabundance estimate** ------------------------
# HT-Seq
htseq.path=file.path(program.dir, "htseq/HTSeq-0.6.1/scripts/htseq-count")
# cufflinks
cufflinks.path=file.path(program.dir, "bin/cufflinks")
```

You can also set the file paths using absolute path (e.g., `fastqc.path="/data/program/bin/fastqc"`).

4.1.2 Software parameters

The default parameters of the external programs are as follows.

4.1.3 Reference files

The reference data used in the pipelines are pre-configured in the file 'data/config.R', which can be customized. If you need to modify it, correct the `config.R` .

```
reference.dir="/data/pubdata"

# Reference FASTA (Version: GRCH38)
ref.fa=file.path(reference.dir,"[[path to referece directory]]/GRCh38_no_alt_analysis_set.201503031.fa"
chrom.fa= file.path(reference.dir, "[[path to referece directory]]/20150407_reference_files/chroms")

# Reference index
- gatk
ref.gold_indels=file.path(reference.dir, "[[path to referece directory]]/Mills_and_1000G_gold_standard.i
ref.dbSNP=file.path(reference.dir, "[[path to referece directory]]/common_all_20161122.vcf")
- bwa
bwa.idx=file.path(reference.dir, "[[path to referece directory]]/bwa_index","GRCH38")
- bowtie2
bowtie.idx=file.path(reference.dir, "[[path to referece directory]]/GRCh38_no_alt_analysis_set.20150303
- STAR
STAR.idx=file.path(reference.dir, "[[path to referece directory]]/gencode.v22")
STAR.idx=file.path(reference.dir, "[[path to referece directory]]/gencode.v27")
- tophat2
transcriptome.idx=file.path(reference.dir, "[[path to referece directory]]/GRCH38")

# Reference GTF,GFF
ref.gtf=file.path(reference.dir, "[[path to referece directory]]/gencode.v27.annotation.gtf")
mir.gff=file.path(reference.dir, "[[path to referece directory]]/hsa.gff3")

# Reference VCF
ref.gold_indels=file.path(reference.dir, "[[path to referece directory]]/Mills_and_1000G_gold_standard.i
ref.dbSNP=file.path(reference.dir, "[[path to referece directory]]/common_all_20161122.vcf")
fn.hapmap.vcf=file.path(reference.dir, "[[path to referece directory]]/resources_broad_hg38_v0_hapmap_3
fn.omni.vcf=file.path(reference.dir, "[[path to referece directory]]/resources_broad_hg38_v0_1000G_omni2
fn.1000g.vcf=file.path(reference.dir, "[[path to referece directory]]/resources_broad_hg38_v0_1000G_phas
cosmic.vcf=file.path(reference.dir, "[[path to referece directory]]/CosmicCodingMuts_v76.vcf")

# Annotation
vep.db.dir=file.path(reference.dir, "[path to VEPdb directory]")
annovar.db.dir=file.path(program.dir, "[path to humandb directory]")
```

4.1.4 Others

The index files used for each step are as follows.
- `fq1.idx` and `fq2.idx` are the index of the fastq file that loads the `qc`, `trim`, `bwa`, `tophat`, `star`, `bowtie2` input file.
- `bam.idx` is the index of the bam file that loads the `rmdu`, `realign`, `gatk`, `varscan`, `muse`, `mutect2`, `somaticsniper` input file.
- `vcf.idx` is the index of the vcf file that loads input file when annotating with `annovar`.

```
fq1.idx=".1.fastq$|_R1.fastq$|.1_val_1.fq$|_1.fq$|.1_val_1.fq$"
fq2.idx=".2.fastq$|_R2.fastq$|.2_val_2.fq$|_2.fq$|.2_val_2.fq$"
bam.idx=".rg.od.bam$|.rmdu.bam$|.realign.bam$|.recal.bam$|.accepted.hits.bam$|.rg.order.bam$|.Aligned.so
vcf.idx=".f.vcf$|_variants.vcf$|.snp.vcf$|.indel.vcf$"
```

If you want to reset the default settings for the index, you can modify it in `SEQprocess/data/config.R`.

5 Configuration of paths and arguments

5.1 Loading package

```
library(SEQprocess)
```

5.2 Set path configuration

Configuration of the installed softwares and data can be managed simply by editing the `data/config.R` file. SEQprocess also supports parallel computation on multi-core machines by using ‘multicore’ R package. When the pipeline is used, `config.R` is automatically run.

5.3 Set output directory

The output directories are basically created `00_qc`, `01_trim`, `02_align`, `03_rmdup`, `04_realign`, `05_vcf`, `06_annot`, `08_cnv`, `09_Robject`. If you run **RSEQ**, `07_RNAquant` folder will be created.

```
  qc.dir=file.path(output.dir, "00_qc"),
  trim.dir=file.path(output.dir, "01_trim"),
  align.dir=file.path(output.dir, "02_align"),
  rmdup.dir=file.path(output.dir, "03_rmdup"),
  realign.dir=file.path(output.dir, "04_realign"),
  vcf.dir=file.path(output.dir, "05_vcf"),
  annot.dir=file.path(output.dir, "06_annot"),
  RNAquant.dir=file.path(output.dir, "07_RNAquant"),
  cnv.dir=file.path(output.dir, "08_cnv"),
  Robject.dir=file.path(output.dir, "09_Robject")
```

5.4 SEQprocess arguments

To run `SEQprocess` pipeline, arguments to be input are required. Following is the description for detailed information.

Arguments name	Type	Default value	Summary
project.name	Character		User set project name which need to make working directory.
fastq.dir	Character		User set input directory stored at fastq files.
output.dir	Character		Set the output directories where the result files will be stored.
config.fn	Character	system.file(“data/config.R”, package = “SEQprocess”)	Configuration of the installed softwares and data can be managed.
type	Character	WGS	Write the type of gene you want to analysis.
pipeline	Character	none	One of the six pipelines provided by SEQprocess.
qc	Logical	TRUE	Quality check of fastq files.
trim.method	Logical	TRUE	Select trimmig off low quality bases.
align.method	Character	bwa	Select align algorithm to use.
bwa.method	Character	mem	Select bwa algorithm to use.
rm.dup	Character	MarkDuplicates	Set the remove duplicates method.
realign	Logical	TRUE	Whether realignment.
variant.call.method	Character	gatk	Set variant call method.

Arguments name	Type	Default value	Summary
annotation.method	Character	annovar	Set variant annotation method.
rseq.abundance.method	Character	none	Set RNA quantification method.
RNAtype	Character	mRNA	Select RNA type when running htseq-count.
report.mode	Logical	FALSE	Wheter to create a report or not.
mc.cores	Integer	1	Set the mc.cores for running multiple samples.

6 Output files

6.1 ExpressionSet

SEQprocess is implemented as a wrapper function in R package, which may facilitate the subsequent analysis under R environment. Furthermore, SEQprocess provides a function to transform the processed data into an 'ExpressionSet' R/bioconductor object, which is a popular data type for the subsequent analyses for biological interpretation. The processed data are transformed into a bioconductor-compatible data type i.e. 'ExpressionSet' which is popularly used for the subsequent analysis NGS data for biological interpretation. We provide these data type files with a filename extension of '.eSet', '.vSet', or 'cSet', for the expression, variants and DNA copy number data, respectively, which may serve a framework to facilitate the subsequent analyses. SEQprocess also provides the ability to convert an ExpressionSet to a SummarizedExperiment data format. Rows of SummarizedExperiment(R/bioconductor) data represent features(e.g. genes, isoforms, exons, proteins, etc…) and columns represent sample data. Like ExpressionSet, SummarizedExperiment is a data type that is easy to analyze after processing.

6.2 Report file

Finally `report.mode` is a function used to combine the inferred samples into one unified table.

In addition, SEQprocess provides a report that summarize the processing steps and visualized tables and plots for the processed results. The report file is automatically generated recording the workflows of the data processing steps, the options used in the processing, and the outcome results.

6.3 Log files

Log files are also generated automatically that can be found in each output directory. Users can find error messages as well as the messages that record each processing step and running times. These will ensure the reproducibility of the data analysis.

7 Usage example

SEQprocess package provides user the function, `SEQprocess`, to run DSEQ and RSEQ pipeline. This function is based on the arguments mentioned above and is run in Rscript.

7.1 Running SEQprocess as a R script for 6 pre-customized pipelines

```
###### dseq-GDC
SEQprocess(project.name="Dseq-GDC-variantcall",
           fastq.dir="[[directory path to rawdata]]",
           output.dir="[[directory path to output]]",
           config.fn="[[directory paht to config]]/config.R",
           type="WES",
           pipeline="GDC",
           variant.call.method="somaticsniper",
           annotation.method = "vep",
           mc.cores=N,
           run.cmd=TRUE )
```

```
###### dseq-GATK
SEQprocess(project.name="Dseq-GATK-variantcall",
           fastq.dir="[[directory path to rawdata]]",
           output.dir="[[directory path to output]]",
           config.fn="[[directory paht to config]]/config.R",
           type="WGS",
           pipeline="GATK",
           mc.cores=N,
           run.cmd=TRUE )
```

```
###### BarSEQ
SEQprocess(project.name="BarSEQ",
           fastq.dir="[[directory path to rawdata]]",
           output.dir="[[directory path to output]]",
           config.fn="[[directory paht to config]]/config.R",
           type="BarSEQ",
           pipeline="BarSEQ",
           mc.cores=N,
           run.cmd=TRUE )
```

```
###### rseq-GDC
SEQprocess(project.name="Rseq-GDC",
           fastq.dir="[[directory path to rawdata]]",
           output.dir="[[directory path to output]]",
           config.fn="[[directory paht to config]]/config.R",
           type="RSEQ",
           pipeline="GDC",
           mc.cores=N,
           run.cmd=TRUE )
```

```
###### rseq-Tuxedo
SEQprocess(project.name="Rseq-Tuxedo",
           fastq.dir="[[directory path to rawdata]]",
           output.dir="[[directory path to output]]",
           config.fn="[[directory paht to config]]/config.R",
           type="RSEQ",
           pipeline="Tuxedo",
           mc.cores=N,
           run.cmd=TRUE )
```

```
###### miR-SEQ
SEQprocess(project.name="miR-SEQ",
           fastq.dir="[[directory path to rawdata]]",
           output.dir="[[directory path to output]]",
           config.fn="[[directory paht to config]]/config.R",
           type="miRSEQ",
           pipeline="miRSEQ",
           mc.cores=N,
           run.cmd=TRUE )
```

7.2 Running SEQprocess as a command line program

SEQprocess can be run as a shell command like a following example.

– Example

```
SEQprocess.R --project.name [Project name]
                     --fastq.dir [Directory path of fastq files]
                     --output.dir [Output directory path]
                     --type ["WGS"/"WES"/"BarSEQ"/"RSEQ"/"miRSEQ"]
                     --pipeline ["none"/"GDC"/"GATK"/"BarSEQ"/"Tuxedo"/"miRSEQ"]
                     --mc.cores [Number of cores]
                     --rum.cmd [Whether run SEQprocess]
                     --qc [T/F]
                     --trim.method ["trim_galore"/"cutadapt"/"none"]
                     --align.method ["bwa"/"bowtie2"/"tophat2"/"star"/"none"]
                     --bwa.method ["mem"/"aln"],
                     --rm.dup ["MarkDuplicates"/"BARCODE"/"none"],
                     --realign [T/F]
                     --variant.call.method ["gatk"/"varscan2"/"mutect2"/"muse"/"somaticsniper"/"none"]
                     --annotation.method ["annovar"/"vep"/"none"]
                     --rseq.abundance.method ["none"/"cufflinks"/"htseq"]
                     --RNAtype["mRNA"/"miRNA"]
                     --cufflinks.gtf ["G"/"g"]
                     --report.mode [T/F]
                     --make.eSet [T/F]
                     --eset2SummarizedExperiment [T/F]
```

Help information is available with `SEQprocess --help`.

8 References

- Backman, T.W.H. and Girke, T. systemPipeR: NGS workflow and report generation environment. BMC Bioinformatics 2016;17(1):388.
- Kwon, S.M., et al. Perspectives of integrative cancer genomics in next generation sequencing era. Genomics Inform 2012;10(2):69-73.
- Schorderet, P. NEAT: a framework for building fully automated NGS pipelines and analyses. BMC Bioinformatics 2016;17:53.
- Singer, J., et al. NGS-pipe: a flexible, easily extendable and highly configurable framework for NGS analysis. Bioinformatics 2018;34(1):107-108.