Rice exome analysis programs - installation and running guide

1) Bioinformatics analysis of whole-exome sequencing datasets

A schematic overview of the bioinformatics pipeline was shown in Figure 1. Analysis-ready BAM files were prepared according to GATK Best Practices for ‘Germline short variant discovery (SNPs + Indels)’ (https://software.broadinstitute.org/gatk/best-practices/workflow?id=11145). Briefly, the clean sequencing reads were mapped to the reference Nipponbare IRGSP-1.0 sequences using the Burrows-Wheeler Aligner (BWA) software (Li and Durbin, 2009), then converted to the standard BAM format, sorted, duplicated reads were marked, realigned near insertions and deletions, and quality scores recalibrated using a combination of Samtools (Li *et al.*, 2009), Picard (Broad Institute), and GATK (McKenna *et al.*, 2010). The following Bash scripts (Lists 1–5) produce an analysis-ready BAM in standard Linux environment.

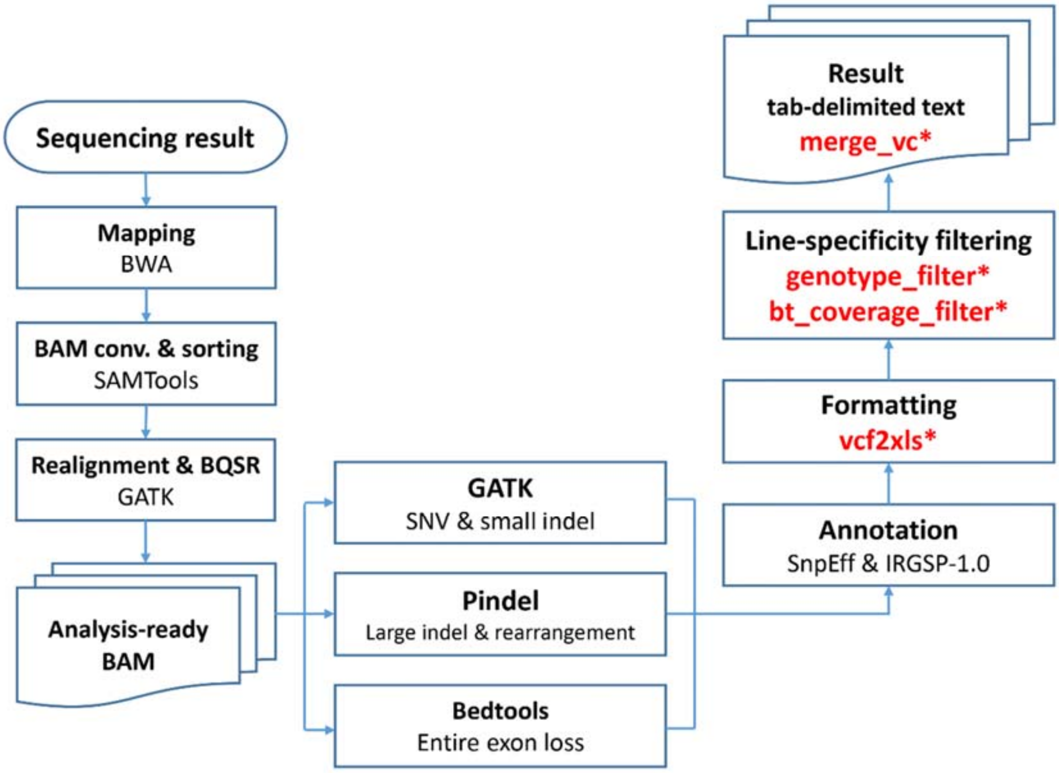

Figure 1. A schematic overview of the bioinformatics pipeline

The first part (left) creates analysis-ready BAM files from cleaned sequencing results. Then variant calling were done with three different algorithms (GATK, Pindel, Bedtools), and the resulting variant candidates were annotated and filtered based on line specificity (see Ichida et al. paper for more details). The programs indicated with asterisks (vcf2xls, genotype_filter, bt_coverage_filter, merge_vc) are provided in this package.

List 1. A typical Bash script for mapping

```
REF=“IRGSP-1.0.fa”
PRJNAME=“project_id”
TARGET=“mutant_name”

BWA_RG_STRING=“-R
\"@RG\\tID:${PRJNAME}\\tPL:Illumina\\tPU:${PRJNAME}_${TARGET}\\tLB:$
{PRJNAME}\\tSM:${TARGET}\""

$BWA mem -t $OMP_NUM_THREADS -v 1 -M -H $REF.dict $BWA_RG_STRING \
  $REF ${TARGET}_1.fq.gz ${TARGET}_2.fq.gz \
  $CMD | $SAMTOOLS view --output-fmt BAM - > $TARGET.bwa.bam
```

List 2. A typical Bash script for sorting

```
$SAMTOOLS sort -@ $OMP_NUM_THREADS -o $TARGET.srt.bam $TARGET.bwa.bam
```

List 3. A typical Bash script for mark duplicates

```
$JAVA -Xmx${MEMORY}m -jar $PICARD MarkDuplicates \
  ASSUME_SORTED=true REMOVE_DUPLICATES=true CREATE_INDEX=true \
  VALIDATION_STRINGENCY=LENIENT \
  I=$TARGET.srt.bam O=$TARGET.rmdup.bam \
  METRICS_FILE=$TARGET.duplicated MAX_RECORDS_IN_RAM=1000000
```

List 4. A typical Bash script for local realignment

```
#Uncomment when processing data from CASAVA version <1.8
#FIXQUAL=“-fixMisencodedQuals”

$JAVA -Xmx${MEMORY}m -jar $GATK -T RealignerTargetCreator \
  $FIXQUAL -nt $OMP_NUM_THREADS -R $REF \
  -I $TARGET.rmdup.bam -o $TARGET.interval_list

$JAVA -Xmx${MEMORY}m -jar $GATK -T IndelRealigner \
  $FIXQUAL -R $REF -targetIntervals $TARGET.interval_list \
  -I $TARGET.rmdup.bam -o $TARGET.realigned.bam
```

List 5. A typical Bash script for base quality score recalibration (BQSR)

```
#Coordinates from rice 3k core SNP v2.1
SNPVCF=“3kcore_snp_2.1.vcf”

$JAVA -Xmx${MEMORY}m -jar $GATK -T BaseRecalibrator \
  -nct $OMP_NUM_THREADS -R $REF -knownSites $SNPVCF \
  -I $TARGET.realigned.bam -o $BAMTEMP/$TARGET$DIVSTR.bqsr

$JAVA -Xmx${MEMORY}m -jar $GATK -T PrintReads \
  -nct $OMP_NUM_THREADS -R $REF -BQSR $TARGET.bqsr \
  -I $TARGET.realigned.bam -o $TARGET.bam

#Recreate BAM indices
rm –f $TARGET.bam.bai $TARGET.bai
$SAMTOOLS index $TARGET.bam
ln –s $TARGET.bai $TARGET.bam.bai
```

Variant callings were done using a combination of the “Unified Genotyper” function in GATK, Pindel (Ye *et al.*, 2009), and Bedtools (Quinlan and Hall, 2010) with the default settings. The called variants were annotated using the SnpEff program (Cingolani *et al.*, 2012) using IRGSP-1.0 annotations (IRGSP-1.0_representative, as of 2015/3/31) as the database. The following Bash scripts (Lists 6–8) produce a tab-delimited list of mutation candidates.

List 6. A typical Bash script for variant calling by GATK

```
BAMSRC=“-I $TARGET1.bam –I $TARGET2.bam”

#Variant calling by GATK
$JAVA -Xmx${MEMORY}m -jar $GATK -T UnifiedGenotyper \
  -nt $OMP_NUM_THREADS -R $REF -glm “BOTH” $GATK_TYPING_OPTION \
  $BAMSRC -o ${PRJNAME}_gatk.vcf

#Filtering
FILTER_STRING=“--clusterWindowSize 10”
$JAVA -Xmx${MEMORY}m -jar $GATK -T VariantFiltration \
  -R $REF $FILTER_STRING -V ${PRJNAME}_gatk.temp.vcf \
  -o ${PRJNAME}_gatk.vcf

#Annotation
$JAVA -Xmx${MEMORY}m -jar $SNPEFF -c snpEff.config -i vcf -o vcf \
  -s ${PRJNAME}_gatk_summary.html ”IRGSP-1.0” \
  ${PRJNAME}_gatk.vcf > ${PRJNAME}_gatk.eff.vcf
```

List 7. A typical Bash script for variant calling by Pindel

```
SCRIPT=“bam_profiles.script”

#Variant calling by Pindel
PINDEL_OPTION=“--window_size 1 --report_long_insertions --
report_breakpoints --minimum_support_for_event 3 --min_inversion_size
10 -T $OMP_NUM_THREADS”
$PINDEL -f $REF -i $SCRIPT $PINDEL_OPTION -o $PRJNAME

#Format conversion
PINDEL2VCF_OPTION=“--gatk_compatible --min_supporting_reads 3”
$PINDEL2VCF $PINDEL2VCF_OPTION -P $PRJNAME -r $REF \
  -R ”IRGSP-1.0” -d ”2015/03/31” -v ${PRJNAME}_pindel.vcf

#Annotation
$JAVA -Xmx${MEMORY}m -jar $SNPEFF -c snpEff.config -i vcf -o vcf \
  -s ${PRJNAME}_pindel_summary.html ”IRGSP-1.0” \
  ${PRJNAME}_pindel.vcf > ${PRJNAME}_pindel.eff.vcf
```

List 8. A typical Bash script for line-specificity filtering and merge results

```
VCF2XLS=“vcf2xls”
VCF2XLS_OPTION=“-a IRGSP-1.0.gff -c \"GT,DP,AD\"”
GTFILTER=“genotype_filter”
TARGETS=“mutant1 mutant2 mutant3”

#GATK
GATK_GTFILTER_OPTION=“-n GATK -a –d”
$VCF2XLS $VCF2XLS_OPTION -i ${PRJNAME}_gatk.eff.vcf $TARGETS \
  > ${PRJNAME}_gatk.xls
$GTFILTER $GATK_GTFILTER_OPTION ${PRJNAME}_gatk.xls \
  > ${PRJNAME}_gatk.line_specific.xls

#Pindel
PINDEL_GTFILTER_OPTION=“-n Pindel -a –d”
$VCF2XLS $VCF2XLS_OPTION -i ${PRJNAME}_pindel.eff.vcf $TARGETS \
  > ${PRJNAME}_pindel.xls
$GTFILTER $PINDEL_GTFILTER_OPTION ${PRJNAME}_pindel.xls \
  > ${PRJNAME}_pindel.line_specific.xls

#Merge results
$MERGE_VC ${PRJNAME}_gatk.line_specific.xls \
  ${PRJNAME}_pindel.line_specific.xls \
  $PRJNAME.line_specific.xls
```

The latest version of the rice exome analysis suite and related auxiliary programs are distributed through our GitHub repository (https://github.com/ion-beam-breeding/RiceExome). The codes can be downloaded by following command line on a typical Linux environment:

```
$ git clone https://github.com/ion-beam-breeding/RiceExome.git
```

2) Programs consisting the package

This package includes `genotype_filter`, a program to determine the line specificity described in the paper, that is designed for GATK and Pindel outputs, and `bt_coverage_filter`, an implementation of line-specificity concept for Bedtools results. Two auxiliary programs, `vcf2xls` and `merge_vc`, are also included: `vcf2xls` converts the standard variant calling format (VCF) to a tab-delimited text, that can easily import to Microsoft Excel or other spreadsheet software, and `merge_vc` combines multiple tab-delimited variant calls produced by `vcf2xls` and outputs non-redundant set of the variations detected in the pipeline.

3) Compile and Installation

The programs provided here was coded in ANSI C++ with some POSIX extensions, and should be possible to compile any standard Linux and UNIX-like environments, including MacOS X. The implementation uses some C++11 (ISO/IEC 14882:2011) extensions, therefore the compiler must support such functionality. These programs are tested on Red Hat Enterprise Linux Server release 7.3 (Maipo) with GNU gcc/g++ 4.8.4, but should be possible to compile and run in any POSIX-compliant environment. In our production environment, Intel C++ Compiler 17.0.4 20170411 was used for compiling both the rice exome programs and the previously published tools.

The compiler-specific parameters (compiler command, optimization flag, etc.) is stored in `Makefile.in`, and included from `Makefile`. The default compiler is GCC and compiled with “-O2 -Wall” options. The followings are the suggested changes to compile with Intel C++ Compiler:

```
CC = icc
CFLAGS = --std=c++11 -O2 –xHOST –Wall –I. ${DEBUG}
CXX = icpc
CXXFLAGS = --std=c++11 -O2 –xHOST –Wall –I. ${DEBUG}
```

To compile, simply enter

```
$ make
```

on your terminal will compile all programs recursively. The executable files are created in *./bin/* directory.

4) Command-line parameters of each program

4.1 vcf2xls

The program **`vcf2xls`** converts the standard variant calling format (VCF) to a tab-delimited text, that can easily import to Microsoft Excel or other spreadsheet software. The **`vcf2xls`** parameters are listed below. Please note that some parameters are mandatory.

Table 1. List and description of vcf2xls parameters

Option	Description
`-i`	The input VCF filename, including relative/full path if necessary. This is a mandatory parameter. It is recommended to use VCF files that were annotated using SnpEff program and IRGSP-1.0 database for full functionality of the program.
`-a`	The input annotation filename. The file should be the standard general feature format (GFF).
`-c`	The columns to export. Specify the VCF keywords listed in the FORMAT field. When multiple FORMAT fields are exported, separate with a comma. To avoid shell unexpected complementation, the parameter values should be quoted with double-quotations.
`-s`	The number of bases from **3' end** that does not allow to use non-ATGC expressions
	Followed by the above options, list the sample names separated with a white space. The sample names must be exact matched with the names listed in the VCF header (case sensitive).

4.2 genotype_filter

The program **`genotype_filter`** provides variant filtering by line-specificity. It uses GATK and Pindel outputs that were previously converted to a tab delimited text by **`vcf2xls`**. The program can also remove low-quality variants by specifying a filtering parameter in the command line. The **`genotype_filter`** parameters are listed below. Please note that some parameters are mandatory.

Table 2. List and description of genotype_filter parameters

Option	Description
`-n`	Program name to show in the first column of the output file. This parameter is to distinguish a line-specific variant was called in which program and does not affect filtering itself. The default value is "Program".
`-e`	The number of heterozygous lines allowed during the line-specificity determination. If this value is set to 1, a variant found in one line (regardless of zygosity) was considered 'line-specific' even if another line has the same variation in heterozygous. Note that if a homozygous variants were found in two or more lines, it will be rejected regardless of this parameter. The defalut value is 0.
`-a`	Add this option to the command line activates alt-rate filter. The alt-rate filter compares the ratio between wild- and mutant-type reads, and determine the ratio is within the expected range from zygosity. In the current implementation, a homozygous variant (either 0/0 or 1/1) is removed from the final output when equal or more than the fraction specified by -f option has 5% or more of another allele (mutant-type allele for 0/0 and wild-type allele for 1/1 variants). The default is off (when -a option is not specified).
`-f`	The maximum allowed fraction of the lines over alt-rate. This parameter changes the threshold to reject a variant by the alt-rate filter. The default value is 0.5.
`-d`	Add this option to the command line activates read-depth filter. The read-depth filter removes low-quality variants, that were covered by sequencing reads below the threshold specified by the -r option. The default is off (when -d option is not specified).
`-r`	The minimum number of reads supporting a variant. This parameter changes the threshold to reject a variant by the read-depth filter. The default value is 10.
`-l`	The maximum number of lines that DOES NOT determined the genotype. Such miss-calling usually happens when read coverage is insufficient. The default value is 0 (which requires all lines has assigned a genotype).
`-v`	Invert the selection. When this option is specified, only variant that were shared between two or more lines will be exported to the output. The default is off (when -v option is not specified).
`-x`	Ignore the specified line from the line-specificity determination. The line name must be exactly match with the names listed in the VCF header. The default value is ''.
	Followed by the above options, list the input filename(s).

4.3 bt_coverage_filter

The program **`bt_coverage_filter`** implements the filtering by line-specificity using Bedtools outputs. Since Bedtools does not determine the genotypes, this program determines whether an uncovered region is specific to one line or not. The **`genotype_filter`** parameters are listed below. Please note that some parameters are mandatory.

Table 3. List and description of bt_coverage_filter parameters

Option	Description
`-n`	Program name to show in the first column of the output file. This parameter is to distinguish a line-specific variant was called in which program and does not affect filtering itself. The default value is “Program”.
`-t`	The number of heterozygous lines allowed during the line-specificity determination. If this value is set to 1, a variant found in one line (regardless of zygosity) was considered ‘line-specific’ even if another line has the same variation in heterozygous. Note that if a homozygous variants were found in two or more lines, it will be rejected regardless of this parameter. The defalut value is 0.
	Followed by the above options, list the input filename(s).

4.4 merge_vc

The program **`merge_vc`** combines multiple tab-delimited variant calls produced by vcf2xls and outputs non-redundant set of the variations detected in the pipeline. To use **`merge_vc`**, simply list the filenames as the command line parameters.

Disclaimer

THIS DOCUMENT IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE DOCUMENT OR THE USE OR OTHER DEALINGS IN THE DOCUMENT.

Copyright ©2018 Hiroyuki Ichida. All rights reserved.

Licensed under Creative Commons license version 4.0 CC BY-NC-SA