RCARE V1.0 Conversion Utilities Manual

SNUBI

Contents

1. Description

RCARE conversion utilities is a set of python-based utilities which convert FASTQ and BAM(binary format for storing sequence data) into VCF(variant call format) and compare RNA to DNA VCF files which are derived from the same sample. The package provides customized TopHat and Samtools command to run. RCARE conversion utilities provide auto installation tools. This makes RCARE conversion utilities execute these functions to use, even for beginners of RNA-seq data analysis.

RCARE conversion utilities contain TopHat (http://tophat.cbcb.umd.edu/), SAMtools(http://samtools.sourceforge.net/),Tabix(http://samtools.sourceforge.net/tabix.shtml), VCFtools(http://vcftools.sourceforge.net/) and bowtie2(http://bowtie-bio.sourceforge.net/bowtie2). If user had pre-set up the tools including TopHat, download the light RCARE conversion utilities and installation.

2. Input data format

The RCARE pre-processing package converts three sequence formats (FASTQ, BAM, VCF) into VCF which is the input format in the RACE web site.

▣ FASTQ format

FASTQ format is a text-based format for storing both the biological sequence (usually nucleotide sequence) and its corresponding quality scores.

```
@SEQ_ID
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
+
!''*((((****))%%%++)(%%%%).1***-+*''))**55CCF>>>>>>CCCCCCC65
```

Fig 1. FASTQ format

▣ BAM format

BAM format is a binary format for storing sequence data. (http://samtools.sourceforge.net/SAMv1.pdf)

▣ VCF format (variant call format)

VCF is a text file format (most likely stored in a compressed manner). It contains meta-information lines, a header line, and data lines with each data line containing information about a position in the genome.

```
##fileformat=VCFv4.1
##fileDate=20090805
##source=myImputationProgramV3.1
##reference=file:///seq/references/1000GenomesPilot-NCBI36.fasta
##contig=<ID=20,length=62435964,assembly=B36,md5=f126cdf8a6e0c7f379d618ff66beb2da,species="Homo sapiens",
##phasing=partial
##INFO=<ID=NS,Number=1,Type=Integer,Description="Number of Samples With Data">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Total Depth">
##INFO=<ID=AF,Number=A,Type=Float,Description="Allele Frequency">
##INFO=<ID=AA,Number=1,Type=String,Description="Ancestral Allele">
##INFO=<ID=DB,Number=0,Type=Flag,Description="dbSNP membership, build 129">
##INFO=<ID=H2,Number=0,Type=Flag,Description="HapMap2 membership">
##FILTER=<ID=q10,Description="Quality below 10">
##FILTER=<ID=s50,Description="Less than 50% of samples have data">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
##FORMAT=<ID=HQ,Number=2,Type=Integer,Description="Haplotype Quality">
#CHROM POS     ID        REF    ALT     QUAL FILTER INFO                              FORMAT      NA00001
20     14370   rs6054257 G      A       29   PASS   NS=3;DP=14;AF=0.5;DB;H2           GT:GQ:DP:HQ 0|0:48:
20     17330   .         T      A       3    q10    NS=3;DP=11;AF=0.017               GT:GQ:DP:HQ 0|0:49:
20     1110696 rs6040355 A      G,T     67   PASS   NS=2;DP=10;AF=0.333,0.667;AA=T;DB GT:GQ:DP:HQ 1|2:21:
20     1230237 .         T      .       47   PASS   NS=3;DP=13;AA=T                   GT:GQ:DP:HQ 0|0:54:
20     1234567 microsat1 GTC    G,GTCT  50   PASS   NS=3;DP=9;AA=G                    GT:GQ:DP    0/1:35:
```

Fig 2. VCF format

3. Installation & Testing the installation

*** install quick start**

▣ RCARE requires pre-setup python environment.

▣ Download RCARE conversion utilities (4.75G) from web site.

▣ Unzip RCARE conversion utilities.

➔ Tar –xvf RCARE_conversion_utilites.tar.gz

▣ Run rcare.py for your purposes.

*** test the installation**

The Sample BAM data contained only 21 chromosomes. These data were extracted from paired-end RNA-seq using the Hela cell in ENCODE (http://genome.ucsc.edu/ENCODE).

▣ Input data confirmation

➔ ls ./input_data/bam/

```
[jjbang@cipher rcare_zip]$ ls ./input_data/bam/
sample.bam
```

▣ Test command

➔ python rcare.py -ib sample.bam -fn sample_bam_test

```
[jjbang@cipher rcare_zip]$ python rcare.py -ib sample.bam -fn sample_bam_test
samtools sort ./input_data/bam/sample.bam ./result_data/bam/sample_bam_test/sorted_sample_bam_test
make RNA VCF file start
samtools mpileup -uf ./resource/bo_index/hg19.fa ./result_data/bam/sample_bam_test/sorted_sample_bam_test.bam | bcftools view -
bvcg -> ./result_data/vcf/sample_bam_test/sample_bam_test.bcf
[mpileup] 1 samples in 1 input files
<mpileup> Set max per-file depth to 8000
[afs] 0:20311.050 1:1463.752 2:1117.198
bcftools view /storage/home/jjbang/rcare_test/rcare_zip/result_data/vcf/sample_bam_test/sample_bam_test.bcf | vcfutils.pl varFi
lter -D500 > /storage/home/jjbang/rcare_test/rcare_zip/result_data/vcf/sample_bam_test/sample_bam_test.vcf
cat /storage/home/jjbang/rcare_test/rcare_zip/result_data/vcf/sample_bam_test/sample_bam_test.vcf | /storage/home/jjbang/rcare_
test/rcare_zip/tools/vcftools_0.1.10/perl/vcf-annotate -f +/Q=20/d=2 > /storage/home/jjbang/rcare_test/rcare_zip/result_data/vc
f/sample_bam_test/sample_bam_test.annotated.vcf
RNA VCF file processing finishing
```

Fig 3. The result of test command

▣ Result confirmation

➔ ls ./result_data/vcf/sample_bam_test/

```
[jjbang@cipher rcare_zip]$ ls ./result_data/vcf/sample_bam_test/
sample_bam_test.annotated.vcf  sample_bam_test.bcf  sample_bam_test.vcf  sample_bam_test_total.vcf
```

4. Synopsis & Example

4.1 Input data folder consists of FASTQ, BAM and VCF.

Insert into row data in each folder.

```
[jjbang@cipher rcare_zip]$ cd input_data/
[jjbang@cipher input_data]$ ls
bam  fastq  vcf
```

Fig 4. Input data format folder

4.2 Convert paired-end FASTQ files into VCF format

➔ python rcare.py -if -p S1.fastq S2.fastq -fn fastq_test

```
[jjbang@cipher rcare_zip]$ python rcare.py -if -p hela1.fastq hela2.fastq -fn fastq_test
tophat --no-novel-juncs -p 8 -G /storage/home/jjbang/rcare_test/rcare_zip/resource/bo_index/hg19.ensembl-for-tophat.gtf  -o /st
orage/home/jjbang/rcare_test/rcare_zip/result_data/bam/fastq_test /storage/home/jjbang/rcare_test/rcare_zip/resource/bo_index/h
g19 /storage/home/jjbang/rcare_test/rcare_zip/input_data/fastq/hela1.fastq /storage/home/jjbang/rcare_test/rcare_zip/input_data
/fastq/hela2.fastq

[2013-09-06 18:35:10] Beginning TopHat run (v2.0.8b)
-----------------------------------------------
[2013-09-06 18:35:10] Checking for Bowtie
                Bowtie version:       2.0.6.0
[2013-09-06 18:35:10] Checking for Samtools
              Samtools version:       0.1.19.0
[2013-09-06 18:35:10] Checking for Bowtie index files
[2013-09-06 18:35:10] Checking for reference FASTA file
[2013-09-06 18:35:10] Generating SAM header for /storage/home/jjbang/rcare_test/rcare_zip/resource/bo_index/hg19
       format:         fastq
       quality scale:  phred33 (default)
[2013-09-06 18:35:16] Reading known junctions from GTF file
[2013-09-06 18:35:36] Preparing reads
```

Fig 5. Example of paired end fastq file converted to VCF using the RCARE conversion utilities

▣ Result confirmation

➔ ls ./input_data/vcf/fastq_test/

```
[jjbang@cipher rcare_zip]$ ls
input_data  lib  rcare.py  resource  result_data  tools
[jjbang@cipher rcare_zip]$ ls result_data/vcf/fastq_test/
fastq_test.annotated.vcf  fastq_test.bcf  fastq_test.vcf  fastq_test_total.vcf
```

4.3 Convert single FASTQ file into VCF format

➔ python rcare.py -if -s S1.fastq -fn single_fastq_test

```
[jjbang@cipher rcare_zip]$ python rcare.py -if -s hela1.fastq  -fn single_fastq_test
tophat --no-novel-juncs -p 8 -G /storage/home/jjbang/rcare_test/rcare_zip/resource/bo_index/hg19.ensembl-for-tophat.gtf  -o /st
orage/home/jjbang/rcare_test/rcare_zip/result_data/bam/single_fastq_test /storage/home/jjbang/rcare_test/rcare_zip/resource/bo_
index/hg19 /storage/home/jjbang/rcare_test/rcare_zip/input_data/fastq/hela1.fastq

[2013-09-07 09:46:41] Beginning TopHat run (v2.0.8b)
-----------------------------------------------
[2013-09-07 09:46:41] Checking for Bowtie
                Bowtie version:       2.0.6.0
[2013-09-07 09:46:42] Checking for Samtools
              Samtools version:       0.1.19.0
[2013-09-07 09:46:42] Checking for Bowtie index files
[2013-09-07 09:46:42] Checking for reference FASTA file
[2013-09-07 09:46:42] Generating SAM header for /storage/home/jjbang/rcare_test/rcare_zip/resource/bo_index/hg19
       format:         fastq
       quality scale:  phred33 (default)
```

Fig 6. Example of single fastq file converted to VCF using the RCARE conversion utilities

▣ Result confirmation

➔ ls ./input_data/vcf/fastq_test

4.4 Convert BAM into convert to VCF format

➔ python rcare.py -ib sample.bam -fn sample_bam_test

```
[jjbang@cipher rcare_zip]$ python rcare.py -ib sample.bam -fn sample_bam_test
samtools sort ./input_data/bam/sample.bam ./result_data/bam/sample_bam_test/sorted_sample_bam_test
make RNA VCF file start
samtools mpileup -uf ./resource/bo_index/hg19.fa ./result_data/bam/sample_bam_test/sorted_sample_bam_test.bam | bcftools view -
bvcg -> ./result_data/vcf/sample_bam_test/sample_bam_test.bcf
[mpileup] 1 samples in 1 input files
<mpileup> Set max per-file depth to 8000
[afs] 0:20311.050 1:1463.752 2:1117.198
bcftools view /storage/home/jjbang/rcare_test/rcare_zip/result_data/vcf/sample_bam_test/sample_bam_test.bcf | vcfutils.pl varFi
lter -D500 > /storage/home/jjbang/rcare_test/rcare_zip/result_data/vcf/sample_bam_test/sample_bam_test.vcf
cat /storage/home/jjbang/rcare_test/rcare_zip/result_data/vcf/sample_bam_test/sample_bam_test.vcf | /storage/home/jjbang/rcare_
test/rcare_zip/tools/vcftools_0.1.10/perl/vcf-annotate -f +/Q=20/d=2 > /storage/home/jjbang/rcare_test/rcare_zip/result_data/vc
f/sample_bam_test/sample_bam_test.annotated.vcf
RNA VCF file processing finishing
```

Fig 7. Example of BAM file converted to VCF using the RCARE conversion utilities

▣ Result confirmation

➔ ls ./result_data/vcf/sample_bam_test

```
[jjbang@cipher rcare_zip]$ ls ./result_data/vcf/sample_bam_test/
sample_bam_test.annotated.vcf  sample_bam_test.bcf  sample_bam_test.vcf  sample_bam_test_total.vcf
```

4.5 Compare RNA vcf with DNA VCF file

➔ python rcare.py -c DNA.vcf RNA.vcf -fn 1_compare_test

```
[jjbang@cipher rcare_zip]$ python rcare.py -c DNA.vcf RNA.vcf -fn 1_compare_test
make[1]: Entering directory `/storage/home/jjbang/rcare_test/rcare_zip/tools/tabix'
gcc -c -g -Wall -O2 -fPIC  -D_FILE_OFFSET_BITS=64 -D_USE_KNETFILE -DBGZF_CACHE  bgzf.c -o bgzf.o
gcc -c -g -Wall -O2 -fPIC  -D_FILE_OFFSET_BITS=64 -D_USE_KNETFILE -DBGZF_CACHE  kstring.c -o kstring.o
gcc -c -g -Wall -O2 -fPIC  -D_FILE_OFFSET_BITS=64 -D_USE_KNETFILE -DBGZF_CACHE  knetfile.c -o knetfile.o
gcc -c -g -Wall -O2 -fPIC  -D_FILE_OFFSET_BITS=64 -D_USE_KNETFILE -DBGZF_CACHE  index.c -o index.o
gcc -c -g -Wall -O2 -fPIC  -D_FILE_OFFSET_BITS=64 -D_USE_KNETFILE -DBGZF_CACHE  bedidx.c -o bedidx.o
ar -csru libtabix.a bgzf.o kstring.o knetfile.o index.o bedidx.o
gcc -c -g -Wall -O2 -fPIC  -D_FILE_OFFSET_BITS=64 -D_USE_KNETFILE -DBGZF_CACHE  main.c -o main.o
gcc -g -Wall -O2 -fPIC  -o tabix main.o -L. -ltabix -lm  -lz
gcc -c -g -Wall -O2 -fPIC  -D_FILE_OFFSET_BITS=64 -D_USE_KNETFILE -DBGZF_CACHE  bgzip.c -o bgzip.o
```

Fig 8. Example of comparing DNA VCF file with RNA VCF file from the same sample source using the RCARE conversion utilities.

▣ Result confirmation

➔ ls ./result_data/compare/1_compare_test/

```
[jjbang@cipher rcare_zip]$ ls ./result_data/compare/1_compare_test/
1_compare_test.vcf
```

4.6 Customized TopHat command running

➔ python rcare.py -tc "tophat" -fn test

```
[jjbang@cipher rcare_zip]$ python rcare.py -tc "tophat" -fn test
tophat:
TopHat maps short sequences from spliced transcripts to whole genomes.

Usage:
    tophat [options] <bowtie_index> <reads1[,reads2,...]> [reads1[,reads2,...]] \
                                    [quals1,[quals2,...]] [quals1[,quals2,...]]
```

Fig9. Example of customized TopHat command running using the RCARE conversion utilities.

4.7 Customized Samtools command running

➔ python rcare.py -sc "samtools" -fn test

```
[jjbang@cipher rcare_zip]$ python rcare.py -sc "samtools" -fn test

Program: samtools (Tools for alignments in the SAM format)
Version: 0.1.19-44428cd

Usage:   samtools <command> [options]
```

Fig10. Example of customized Samtools command running using the RCARE conversion utilities.

5. RCARE pre-processing options

option	description
-if	input file format: FASTQ file
-ib	input file format : BAM file
-p	paired end FASTQ file
-c	Compare VCF(RNA) with VCF(DNA)
-fn	Result file name

-tc	Customized tophat commands
-sc	Customized samtools commands

6. Package composition

Folder file name	Description
input_data	Insert input data
result_data	Save result data
resource	Required files for preprocessing
tools	Required tools for preprocessing
rcare.py	batch file of conversion utilities

▣ RCARE conversion utilities needs pre-installation of python.

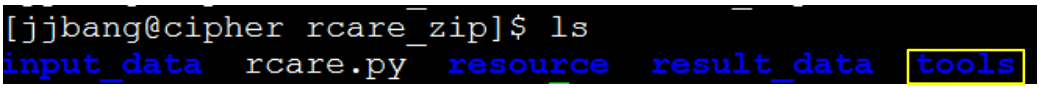
7. Light RCARE conversion utilities installation

▣ Download light-RCARE-conversion_utilites.tar.gz (35.62MB) from RCARE web site.

▣ Download tools

- TopHat : http://tophat.cbcb.umd.edu/
- Samtools : http://samtools.sourceforge.net/
- Bowtie2 : http://bowtie-bio.sourceforge.net/bowtie2/index.shtml
- Tabix : http://samtools.sourceforge.net/tabix.shtml
- Vcftools : http://vcftools.sourceforge.net/

▣ User should insert all tools into the tools folder of the RCARE conversion utilities.



▣ If user used previous setup tools, user zeroize each tool's environment settings.

8. Authors

Ju Han Kim and Soo Youn lee from SNUBI (Seoul National University Biomedical Informatics; http://www.snubi.org/)

9. Reference

1. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R. The Sequence Alignment/Map format and SAMtools. Bioinformatics 2009; 25: 2078-9.

2. Trapnell C, Pachter L, Salzberg SL. TopHat: discovering splice junctions with RNA-Seq. Bioinformatics 2009; 25: 1105-11.

3. Danecek P, Auton A, Abecasis G, Albers CA, Banks E, DePristo MA, Handsaker RE, Lunter G, Marth GT, Sherry ST, McVean G, Durbin R. The variant call format and VCFtools. Bioinformatics 2011; 27: 2156-8.

4. Langmead B, Salzberg SL. Fast gapped-read alignment with Bowtie 2. Nature methods 2012; 9: 357-9.

5. Li H. Tabix: fast retrieval of sequence features from generic TAB-delimited files. Bioinformatics 2011; 27: 718-9.