MutMap pipeline quick start guide (for framework ver 1.4.4)

Version 1.4.4

Rev.0.0

2.July.2014

0. Preparation

0.1. Prerequisites

1. Perl (v5.8.8)
2. R (version 2.15.0)
3. BWA (version 0.5.9-r16)
4. SAMtools (0.1.8 or before)
5. FASTX-Toolkit (http://hannonlab.cshl.edu/fastx_toolkit/)

0.2. Requirement

1. public genome sequence (fasta)
   (For rice, Nipponbare reference genome is used.)

2. your sample's fastq files

   - fastq files (archived in gzip) to make "your reference"
   - your bulked sample's fastq files (archived in gzip)

In this guide, it is expected that

ID names in public fasta don't have illegal characters
(e.g., '|'=pipe, ' '=space, …), and
there are no blank lines, and
base data are word-wrapped regularly.

Those fastqs follow Sanger FASTQ quality score format
(illumina CASAVA1.8 later)

1.Install pipeline to your system(Linux)

```
$ cd /myhome                                        (#) replace "/myhome" with the proper path
$ cp MutMap_framework1.4.4.tar.gz .                 name according to your system environment

$ tar zxvf MutMap_framework1.4.4.tar.gz             (#) decompression for tar.gz

$ mv MutMap_framework1.4.4 MutMap_test              (#) renamed ; new name is
                                                    "MutMap_test" in here
```

2.Provide your sample's fastq files to each directory in this pipeline

```
$ ls /fastq/ZZZ/
ZZZ_L003_R1_001.fastq.gz                    (#) e.g., Your fastq files to make your
ZZZ_L003_R2_001.fastq.gz                    reference fasta are stored in "/fastq/ZZZ"
ZZZ_L004_R1_001.fastq.gz                    directory
ZZZ_L004_R2_001.fastq.gz

$ ls /fastq/XXX/
XXX_L005_R1_001.fastq.gz                    (#) e.g., Your bulked sample's fastq files are
XXX_L005_R2_001.fastq.gz                    stored in "/fastq/XXX" directory
XXX_L006_R1_001.fastq.gz
XXX_L006_R2_001.fastq.gz


$ cd /myhome/MutMap_test/1.qualify_read/anyname
$ ln -s /fastq/ZZZ/ZZZ_L003_R1_001.fastq.gz p_3_1_sequence.txt.gz
$ ln -s /fastq/ZZZ/ZZZ_L003_R2_001.fastq.gz p_3_2_sequence.txt.gz
$ ln -s /fastq/ZZZ/ZZZ_L004_R1_001.fastq.gz p_4_1_sequence.txt.gz
$ ln -s /fastq/ZZZ/ZZZ_L004_R2_001.fastq.gz p_4_2_sequence.txt.gz

$ cd /myhome/MutMap_test/1.qualify_read/mybulk
$ ln -s /fastq/XXX/XXX_L005_R1_001.fastq.gz p_5_1_sequence.txt.gz
$ ln -s /fastq/XXX/XXX_L005_R2_001.fastq.gz p_5_2_sequence.txt.gz
$ ln -s /fastq/XXX/XXX_L006_R1_001.fastq.gz p_6_1_sequence.txt.gz
$ ln -s /fastq/XXX/XXX_L006_R2_001.fastq.gz p_6_2_sequence.txt.gz
```

(#) provide your fastq files with a new name according to the naming convention by link command "ln –s".

Naming convention

*_[0-9]*_[12]_sequence.txt.gz

[0-9]* : unique number; usually some number derived from lane number of flow cell
[12] : must be assigned 1 or 2, which means "1st" or "2nd" of a pair-end read

3. Provide fasta (downloaded fasta) to “downloaded_fasta” in this pipeline

```
$ cp public.fasta /myhome/MutMap_test/downloaded_fasta/
```

You copy “public.fasta” which is downloaded fasta file
to “downloaded_fasta” directory in this pipeline, as shown in above.

4.Setup “common.fnc”

4.1. Edit “config.txt” in top directory of this pipeline

before line 12, 15, 31, 115

```
...
Key1_Bulked_sample_name="mybulk"

Key1_My_cultivar_sample="anyname"
...
Key2_Path_public_reference_FASTA="./downloaded_fasta/IRGSP1.0_genome..."
...
Key3_Individuals=20
```

after

```
...
Key1_Bulked_sample_name="XXX"

Key1_My_cultivar_sample ="ZZZ"
...
Key2_Path_public_reference_FASTA="./downloaded_fasta/public.fasta"
...
Key3_Individuals=25
```

(#) ”Key3_Individual” means number of individual of your bulked sample

You keep other Key paramers as default value(no change) usually

4.2. Run “Bat_make_common.fnc.sh”

```
$ cd /myhome/MutMap_test

$ ./Bat_make_common.fnc.sh

mv 1.qualify_read/mybulk 1.qualify_read/XXX
mv 1.qualify_read/anyname 1.qualify_read/ZZZ
mkdir -p 1.qualify_read/XXX/q30p90/sep_pair
mkdir -p 1.qualify_read/ZZZ/q30p90/sep_pair
```

5.1. [Step 1] Qualify fastq files

5.1.1. Run “Run_all_Bats.sh” with an argument [9|0]

```
$ cd /myhome/MutMap_test/1.qualify_read

$ ./Run_all_Bats.sh 9
```

after that, you get qualified fastq files from your “ZZZ” fastq files to make “your reference” fasta

```
$ cd /myhome/MutMap_test/1.qualify_read

$ ./Run_all_Bats.sh 0
```

after that, you get qualified fastq files from your bulked sample “XXX” fastq files

5.2. [Step2] Make your reference fasta

5.2.1. Run “Run_all_Bats.sh”

```
$ cd /myhome/MutMap_test/2.make_consensus/

$ ./Run_all_Bats.sh
```

after that, you get “your reference” fasta.

5.2.2. Run “Run_all_Bats.sh”

```
$ cd /myhome/MutMap_test/2.make_consensus/90.align_to_this_fasta/

$ ./Run_all_Bats.sh
```

after that, you get false-positive SNPs which may disturb your analysis. in next steps, those SNPs will be rejected.

5.3. [Step3] Align your bulked sample's fastq to "your reference" fasta and main analysis

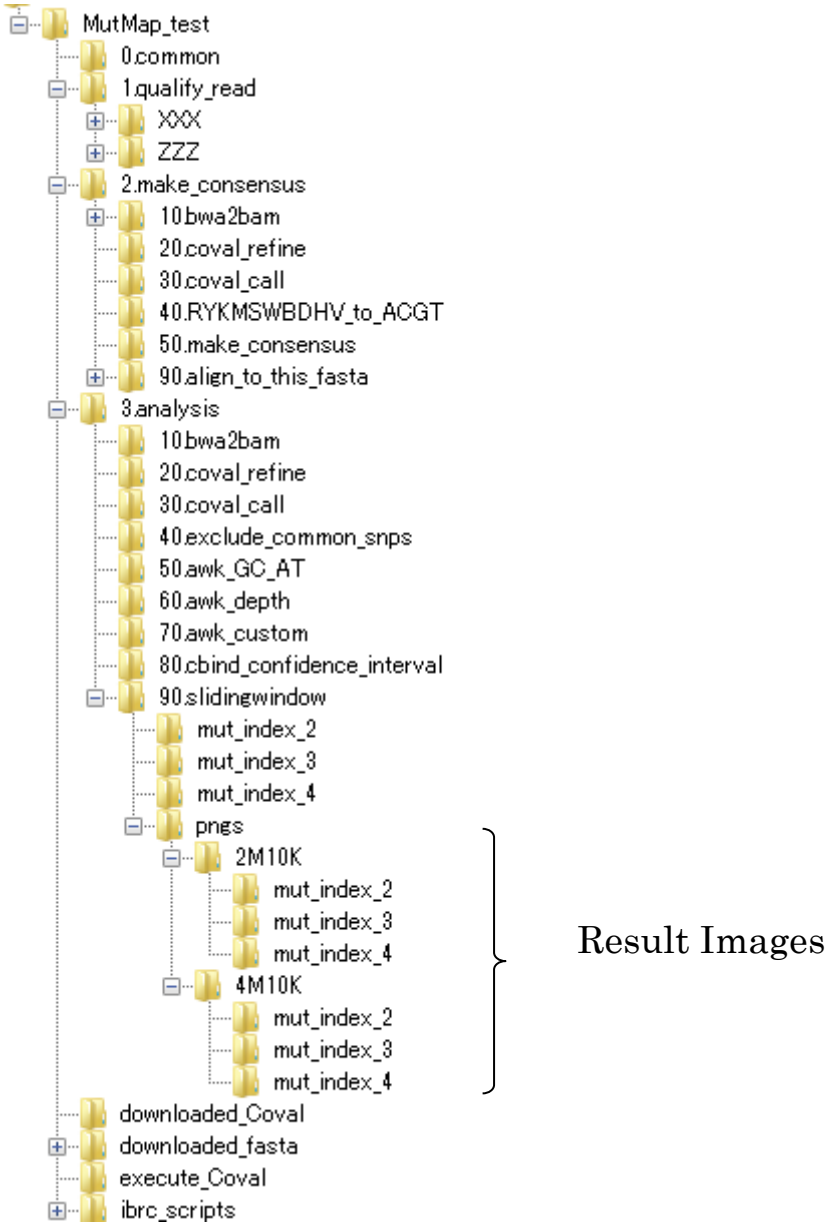
5.3.1. Run "Run_all_Bats.sh"

```
$ cd /myhome/MutMap_test/3.analysis/

$ ./Run_all_Bats.sh
```

after that, you get SNPs result of your bulked sample, via
bwa alignment,
coval filtering and some kind of filtering,
calculation of confidence intervals,
sliding window smoothing and visualization of numeric data,
finally you get result as below images(png)

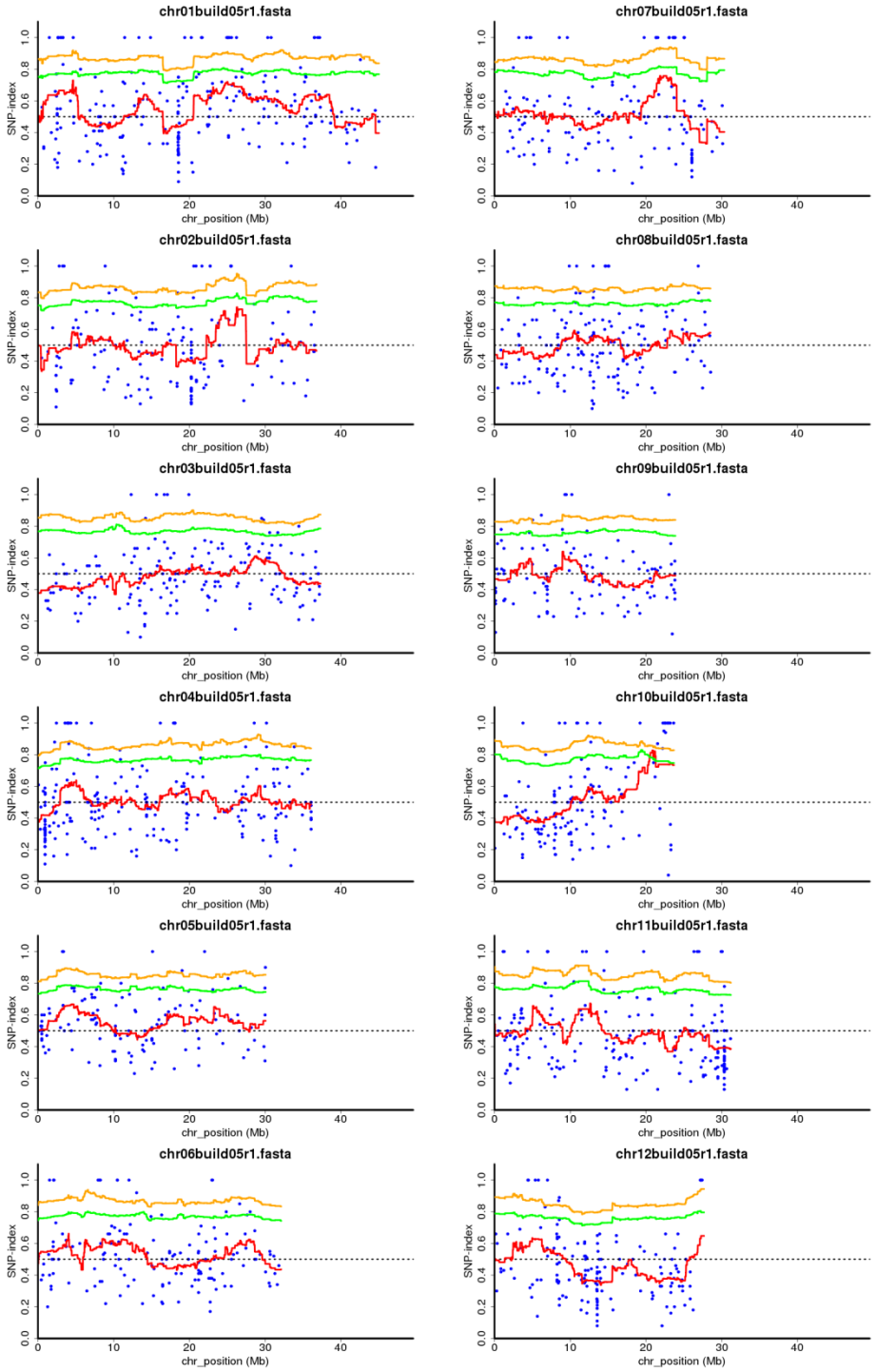
6. Directories



7. Result Images

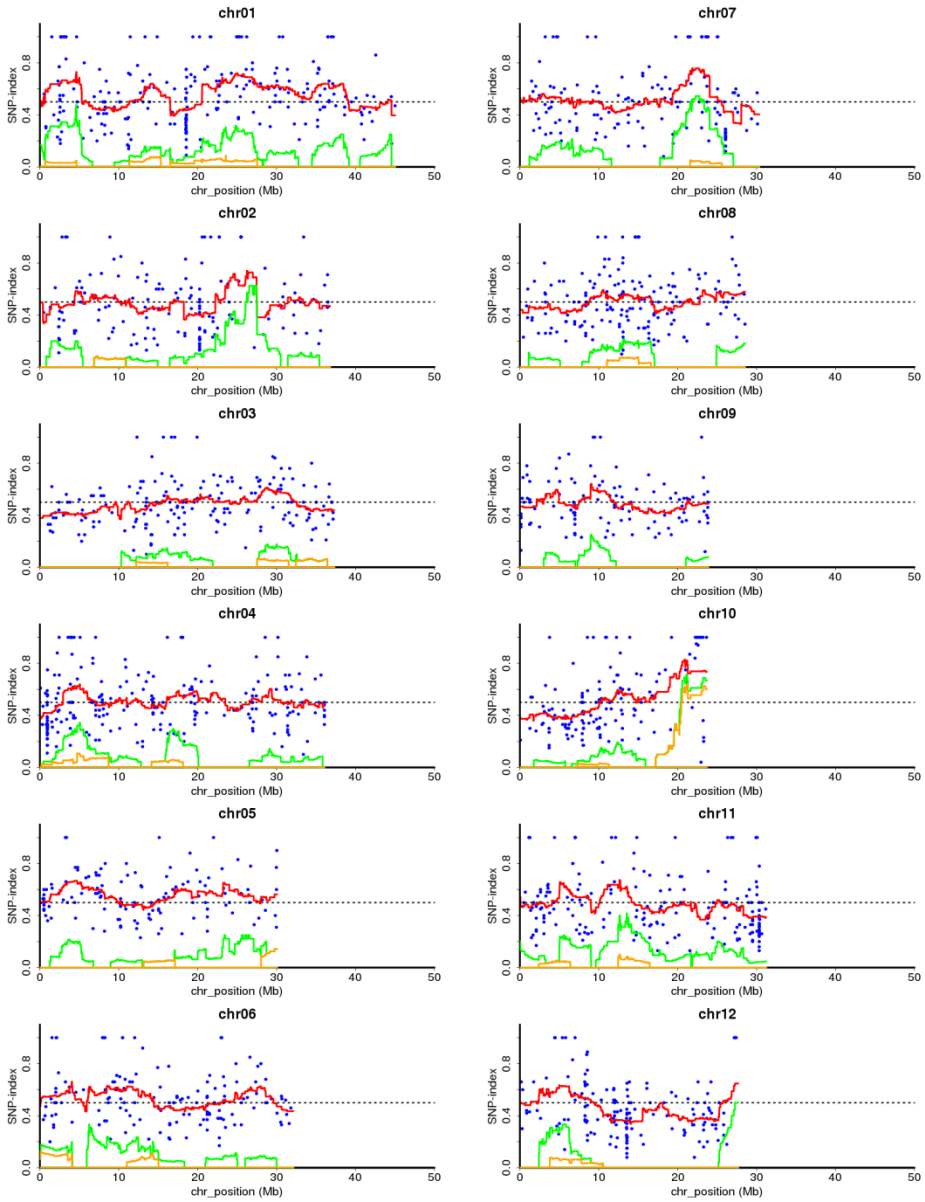
7.1. SNP-index

e.g., “XXX_q30p90_filtered_pvalue_sldwnd2M10K_cov2_co5.png”



Blue dot : SNP-index
Red line : sliding window average for SNP-index
Green line : sliding window average for 95%-confidence interval
Orange line : sliding window average for 99%-confidence interval

7.2. SNP-index with additional curve,
e.g., “ratio_XXX_q30p90_filtered_pvalue_sldwnd2M10K_cov2_co5.png”



Blue dot : SNP-index
Red line : sliding window average for SNP-index
Green line : per window count of SNP which gives SNP-index value > 95%confidence interval
Orange Line : per window count of SNP which gives SNP-index value > 99%-confidence interval