**GradingPoolSeq——pipeline**

**Prerequisites**

•Install the following programs:

1. Perl (v5.22.1)

2. Perl module (Statistics::Distributions)

3.R(v3.2.5)

4.R package (ggplot2)

Based on linux system.

Installation of perl: <https://www.perl.org/get.html>

Installation of R: <https://www.r-project.org/>

Which would intall within one hour.

None required non-standard hardware.

**Demo :**

Demo is in demo file, including instructions and expected outputs. Expected run time is within 20 minutes.

**Pipeline:**

Step 1: Install this pipeline framework to your system.

ex) $ cd /GradingPool *#Working place*

$ cp GradingPool\_framework.tar.gz

$ tar zxvf GradingPool\_framework.tar.gz

Step 2: Prepare each bulks and reference genome sequencing data and perform alignment as GATK “best practice” to obtain the results of variant calling.

Step 3: Undertake filtering process

Suppose we have six bulks here, four of F2 generations (pool1-pool4) and two of parental lines (pool5 and pool6). If you want to scrutiny the results of the combination of two or three pools, you can change the parameter where I point out specifically.

1. filter out low-quality variants

*#input file: Headingdate .vcf*

*#output file: filter1.txt*

*#100: quality threshold we set here as an example.*

perl Filter\_Quality.pl Headingdate .vcf > filter1.txt 100

1. select variants with appropriate depth(5%~95%)

*#input file: filter1.txt*

*#output file: filter2.txt*

*#6: total pools (including parents’ pools)*

*#0.05&0.95: depth range from 5% to 95%.*

perl Filter\_Depth.pl filter1.txt > filter2.txt 6 0.05 0.95

1. screen out variants that both parental lines present homogeneous and different genotypes(optional)

*#input file: filter2.txt*

*#output file: filter3.txt*

*#5&6: numerical order of two parents’ pool*

perl Filter\_Parent.pl filter2.txt > filter3.txt 5 6

1. filter out SNPs that show none reference reads.

*#input file: filter3.txt*

*#output file: filter4.txt*

*#6: total pools (including parents’ pools)*

*#1,2,3,4: numerical order of second generations ’ pool.* *If your want to find out the result of other combinations, for example, the first pool and the fourth pool, the parameter will be 1 and 4, and so on.(perl Filter\_SNP.pl filter3.txt > filter4.txt 6 1 4)*

perl Filter\_SNP.pl filter3.txt > filter4.txt 6 1 2 3 4

Step 4: calculate p-value

*#input file: filter4.txt*

*#output file: pvalue.txt*

*#1,2,3,4: numerical order of second generations ’ pool. If your want to find out the result of other combinations, for example, the first pool and the fourth pool, the parameter will be 1 and 4, and so on.(perl Ridit.pl filter4.txt > pvalue.txt 1 4)*

perl Ridit.pl filter4.txt > pvalue.txt 1 2 3 4

Step 5: denoise strategy

*#input file: pvalue.txt*

*#output file: ratio.txt*

*#2: set 2(X100kb) as a defined genomic interval (200kb in each chromosome window)*

*#10: threshold for high significant variants, generally we set 10.*

*#12: the number of chromosome*

perl denoise.pl pvalue.txt > ratio.txt 2 10 12

Step 6: graph analysis

First install packages in R:

1. generate P-value plot

*#input file: pvalue.txt*

*#output file: pvalue.png*

Rscript pos-pvalue.R

1. generate ratio plot

*#input file: ratio.txt*

*#output file: ratio.png*

Rscript pos-ratio.R

Step 7: identify peak interval

*#input file: ratio.txt*

*#1: chromosome*

*#output: peak point and peak interval*

perl p-max.pl ratio.txt 1