

qcCHIP User's Guide

Xiang Liu xiang.liu@moffitt.org
Mingxiang Teng mingxiang.teng@moffitt.org
Department of Biostatistics and Bioinformatics
Moffit Cancer Center, Tampa, FL, USA

2025-04-28

Contents

Introduction	1
Install package	1
Getting Started	2
Preparing Input files	2
Using VCF files as input	4
Basic Usage of <i>qcCHIP</i>	8
Basic Usage of <i>CHIPfilter</i>	12

Introduction

Clonal hematopoiesis (CH) is a molecular biomarker associated with various adverse outcomes in healthy and disease individuals. Detecting CHs usually involves genomic sequencing of individual blood samples followed by rigorous bioinformatics data filtering. We report an R package *qcCHIP*, a bioinformatics pipeline to identify CH from sequencing data by implementing a series of quality control filters and a permutation-based parameter optimization.

Install package

Install *qcCHIP* package via *devtools*.

```
library(devtools)
devtools::install_github("https://github.com/tenglab/qcCHIP.git",force=T)
#> -- R CMD build -----
#>    checking for file '/private/var/folders/3s/9r3z6h_n0z3889fx5pn38svw0021h2/T/Rtmpu0xu/remotes2
#>    - preparing 'qcCHIP':
```

```
#>      checking DESCRIPTION meta-information ... v checking DESCRIPTION meta-information
#> - checking for LF line-endings in source and make files and shell scripts
#> - checking for empty or unneeded directories
#> - building 'qcCHIP_0.0.0.9000.tar.gz'
#>      Warning: invalid uid value replaced by that for user 'nobody'
#>
#>
```

Getting Started

Load the package in R.

```
library(qcCHIP)
library(GenomicRanges)
```

Preparing Input files

qcCHIP requires an annotated text file with specific column names:

All empty value should be noted as “.”.

1. Chr: chromosome of variant. Exp: chr1, chr2,chrX.
2. Start: start position of variant.
3. End: end position of variant.
4. Ref: reference allele.
5. Alt: alternative allele.
6. TLOD: TLOD or Qual Info from vcf file.
7. SOR: SOR Info from vcf file.
8. AD_alt: Allelic depths for the alt alleles from vcf file.
9. AF: AF or VAF from blood sample vcf file.
10. DP: DP from vcf file.
11. SAF: SAF info from vcf file.
12. SAR: SAR info from vcf file.
13. SampleID: sample ID or variant.
14. Func.refGene: function annotation from refGene.
15. ExonicFunc.refGene: exonic function annotation from refGene. (nonsynonymous SNV and synonymous SNV values need to be named as “nonsynonymous SNV” and”synonymous SNV”)
16. cosmic70: if the variant is exist in cosmic database. (empty value needs to be “.”)
17. tumor_AF: optional, AF or VAF from tumor sample vcf file.
18. non_cancer_AF_popmax: optional, non cancer AF value from gnomad database.
19. Alt_dpGAP_PopFreq: optional, ALT population frequency from dpGAP databse.

There are two ways to prepare the input file.

1. Using in-house converting tool to convert raw VCF or annotated VCF files.
2. Manually create the input file with the columns described above.


```
#> 1      .      2.03E-05      .      .      .
#> 2      .      .      .      .      .
#> 3      1      0.6222      .      .      .
#> 4      .      .      .      .      .
#> 5      .      .      .      .      .
#> 6      .      .      .      .      .
```

Using VCF files as input

Tool vcf2input can convert raw or annotated VCF files to the input file that can be used for qcCHIP.

Raw VCF files

Below are the examples of how to convert raw VCF files to input file.

```
# Path of vcf files
raw_vcf_path<- system.file("extdata/raw_vcf",package="qcCHIP")

# Create data.frame of samples
sample_df <- data.frame(blood=list.files(raw_vcf_path)[! list.files(raw_vcf_path) %in%
                                                                grep("*_tumor\\.vcf\\.gz",list.files(raw_vcf_p
                                                                tumor=grep("*_tumor\\.vcf\\.gz",list.files(raw_vcf_path),value=T))

head(sample_df)
#>      blood      tumor
#> 1 sample_1.vcf.gz sample_1_tumor.vcf.gz
#> 2 sample_10.vcf.gz sample_10_tumor.vcf.gz
#> 3 sample_2.vcf.gz sample_2_tumor.vcf.gz
#> 4 sample_3.vcf.gz sample_3_tumor.vcf.gz
#> 5 sample_4.vcf.gz sample_4_tumor.vcf.gz
#> 6 sample_5.vcf.gz sample_5_tumor.vcf.gz

# 1. No paired tumor sample and no annotation of other database
input_1 <- vcf2input(vcf_path = raw_vcf_path,
                    sample_list=sample_df$blood)
head(input_1)
#>      SampleID Chr      Start      End Ref Alt      TLOD      SOR AD_alt      AF      DP
#> 1 sample_1_st_g chr1 1786850 1786850 G T 4.08 0.375 3 0.103 30
#> 2 sample_1_st_g chr1 1786996 1786996 C G 43.36 0.333 17 0.469 36
#> 3 sample_1_st_g chr1 36471458 36471458 A G 187.69 0.43 72 0.500 148
#> 4 sample_1_st_g chr1 64833405 64833405 A C 18.22 0.593 8 0.532 14
#> 5 sample_1_st_g chr1 64837976 64837976 C T 14.64 1.244 7 0.269 30
#> 6 sample_1_st_g chr1 64841324 64841324 G T 3.08 1.142 3 0.062 76
#> SAF SAR
#> 1 2 1
#> 2 9 8
#> 3 36 36
#> 4 3 5
#> 5 5 2
#> 6 1 2

# 2. With paired tumor sample and no annotation of other database
input_2 <- vcf2input(vcf_path = raw_vcf_path,
```

```

sample_list=sample_df$blood,
tumor=T,
tumor_path=raw_vcf_path,
tumor_list=sample_df$tumor)

head(input_2)
#>   Chr   Start Ref Alt   SampleID   End   TLOD   SOR AD_alt   AF   DP
#> 1 chr1 1786850   G   T sample_1_st_g 1786850 4.08 0.375    3 0.103 30
#> 2 chr1 1786996   C   G sample_1_st_g 1786996 43.36 0.333   17 0.469 36
#> 3 chr1 36471458  A   G sample_1_st_g 36471458 187.69 0.43    72 0.500 148
#> 4 chr1 64833405  A   C sample_1_st_g 64833405 18.22 0.593    8 0.532 14
#> 5 chr1 64837976  C   T sample_1_st_g 64837976 14.64 1.244    7 0.269 30
#> 6 chr1 64841324  G   T sample_1_st_g 64841324 3.08 1.142    3 0.062 76
#>   SAF SAR   SampleID_t tumor_AF
#> 1   2   1           .           .
#> 2   9   8 sample_1_st_t 0.216
#> 3  36  36 sample_1_st_t 0.16
#> 4   3   5 sample_1_st_t 0.534
#> 5   5   2 sample_1_st_t 0.483
#> 6   1   2           .           .

```

Note that, raw vcf files don't have the required information of Func.refGene, ExonicFunc.refGene, and cosmic70 for qcCHIP. Users need to manually adding the corresponding columns to the resulting table converted from raw VCF.

Annotated VCF files

Currently, qcCHIP can take the following annotation information.

1. Function name, gene name, exonic function, AAchange, and GeneDetail information from refGene database;
2. COSMIC info from COSMIC database (e.g. COSMIC70);
3. Non cancer AF population max from gnomad database.

Function name, exonic function, and COSMIC are required annotation information for qcCHIP.

```

# Path of vcf files
annot_vcf_path<- system.file("extdata/annot_vcf",package="qcCHIP")

# Create data.frame of samples
sample_df <- data.frame(blood=list.files(annot_vcf_path)[! list.files(annot_vcf_path) %in%
                                                                grep("*_tumor\\.hg38_multianno\\.vcf",list.files(
                                                                tumor=grep("*_tumor\\.hg38_multianno\\.vcf",list.files(annot_vcf_path),value=T))

head(sample_df)
#>           blood           tumor
#> 1 sample_1.hg38_multianno.vcf sample_1_tumor.hg38_multianno.vcf
#> 2 sample_10.hg38_multianno.vcf sample_10_tumor.hg38_multianno.vcf
#> 3 sample_2.hg38_multianno.vcf sample_2_tumor.hg38_multianno.vcf
#> 4 sample_3.hg38_multianno.vcf sample_3_tumor.hg38_multianno.vcf
#> 5 sample_4.hg38_multianno.vcf sample_4_tumor.hg38_multianno.vcf
#> 6 sample_5.hg38_multianno.vcf sample_5_tumor.hg38_multianno.vcf

# example of annotated VCF files

```

```

library(vcfR)
exp_vcf <- read.vcfR(system.file("extdata/annot_vcf", "sample_1.hg38_multianno.vcf", package="qcCHIP"))
#> Scanning file to determine attributes.
#> File attributes:
#>   meta lines: 3468
#>   header_line: 3469
#>   variant count: 213
#>   column count: 10
#> Meta line 1000 read in.Meta line 2000 read in.Meta line 3000 read in.Meta line 3468 read in.
#> All meta lines processed.
#> gt matrix initialized.
#> Character matrix gt created.
#>   Character matrix gt rows: 213
#>   Character matrix gt cols: 10
#>   skip: 0
#>   nrows: 213
#>   row_num: 0
#> Processed variant: 213
#> All variants processed

# check name of annotation information
strsplit(head(exp_vcf@fix,n=1)[,8],";")
#> $INFO
#> [1] "AS_SB_TABLE=19,8/2,1"
#> [2] "DP=32"
#> [3] "ECNT=1"
#> [4] "FS=0"
#> [5] "MBQ=30,20"
#> [6] "MFRL=182,144"
#> [7] "MMQ=60,60"
#> [8] "MPOS=10"
#> [9] "POPAF=7.3"
#> [10] "SOR=0.375"
#> [11] "TLOD=4.08"
#> [12] "ANNOVAR_DATE=2020-06-08"
#> [13] "Func.refGeneWithVer=UTR3"
#> [14] "Gene.refGeneWithVer=GNB1"
#> [15] "GeneDetail.refGeneWithVer=NM_001282538.1:c.*213C>A\|x3bNM_001282539.1:c.*213C>A\|x3bNM_002074.1:c.*213C>A"
#> [16] "ExonicFunc.refGeneWithVer=."
#> [17] "AAChange.refGeneWithVer=."
#> [18] "cosmic70=."
#> [19] "AF=."
#> [20] "AF_popmax=."
#> [21] "AF_male=."
#> [22] "AF_female=."
#> [23] "AF_raw=."
#> [24] "AF_afr=."
#> [25] "AF_sas=."
#> [26] "AF_amr=."
#> [27] "AF_eas=."
#> [28] "AF_nfe=."
#> [29] "AF_fin=."
#> [30] "AF_asj=."

```

```

#> [31] "AF_oth=."
#> [32] "non_topmed_AF_popmax=."
#> [33] "non_neuro_AF_popmax=."
#> [34] "non_cancer_AF_popmax=."
#> [35] "controls_AF_popmax=."
#> [36] "ALLELE_END"

# Annotated VCF of sample and paired tumor sample. Annotated with refGene, cosmic, and gnomad.
input_3 <- vcf2input(vcf_path = annot_vcf_path,
                    sample_list=sample_df$blood,
                    tumor=T,
                    tumor_path=annot_vcf_path,
                    tumor_list=sample_df$tumor,
                    refGene=T,
                    refGene_func_name="Func.refGeneWithVer",
                    refGene_gene_name="Gene.refGeneWithVer",
                    refGene_Exonicfunc_name="ExonicFunc.refGeneWithVer",
                    refGene_AAchange_name=NA,
                    refGene_GeneDetail_name=NA,
                    cosmic=T,
                    cosmic_name="cosmic70",
                    gnomad=T,
                    gnomad_name="non_cancer_AF_popmax")

head(input_3)
#>   Chr   Start Ref Alt   SampleID   End   TLOD   SOR AD_alt   AF   DP
#> 1 chr1 1786850   G   T sample_1_st_g 1786850 4.08 0.375     3 0.103 30
#> 2 chr1 1786996   C   G sample_1_st_g 1786996 43.36 0.333    17 0.469 36
#> 3 chr1 36471458  A   G sample_1_st_g 36471458 187.69 0.43     72 0.500 148
#> 4 chr1 64833405  A   C sample_1_st_g 64833405 18.22 0.593     8 0.532 14
#> 5 chr1 64837976  C   T sample_1_st_g 64837976 14.64 1.244     7 0.269 30
#> 6 chr1 64841324  G   T sample_1_st_g 64841324 3.08 1.142     3 0.062 76
#>   SAF SAR   SampleID_t tumor_AF Func.refGene Gene.refGene ExonicFunc.refGene
#> 1   2   1           .           .          UTR3          GNB1                .
#> 2   9   8 sample_1_st_t   0.216          UTR3          GNB1                .
#> 3  36  36 sample_1_st_t   0.16          exonic        CSF3R      synonymous_SNV
#> 4   3   5 sample_1_st_t   0.534          UTR3          JAK1                .
#> 5   5   2 sample_1_st_t   0.483          exonic        JAK1      synonymous_SNV
#> 6   1   2           .           .          exonic        JAK1      stopgain
#>   cosmic70 non_cancer_AF_popmax Alt_dpGAP_PopFreq
#> 1           .           .
#> 2           .           .
#> 3           .       0.7273           .
#> 4           .           .
#> 5           .       0.1083           .
#> 6           .           .

```

This output table is the correct format that can be directly used to find CH.

```

# find CH with default metrics
ch_candidate <- CHIPfilter(input_3,tumor_sample=T)
#> [1] "Perform population metrics"
#> [1] "Perform technique metrics"
#> [1] "Perform individual metrics with paired tumor sample"

```

```

#> [1] "Perform functional metrics"
#> [1] "Perform not nonsunonymous metrics"
#> [1] "Perform gnomad metrics only"
#> [1] "No blacklist region bed file find, skip"
head(ch_candidate)
#>      Chr      Start Ref Alt      SampleID      End      TLOD      SOR AD_alt      AF
#> 347 chr5 150068242 GAGC      G sample_10_st_g 150068245 18.15 1.508      8 0.036
#> 391 chrX 48792153      G      C sample_10_st_g 48792153 205.62 0.716      85 0.296
#> 2170 chr7 102255385      C CAA sample_9_st_g 102255385 31.03 0.284      49 0.187
#> 2176 chr7 102257898      C CT sample_9_st_g 102257898 14.91 0.565      19 0.233
#>      DP SAF SAR      SampleID_t tumor_AF Func.refGene Gene.refGene
#> 347 291 6 2 sample_10_st_t 0.013      exonic      SETBP1
#> 391 299 40 45 sample_10_st_t 0.026      exonic      BCORL1
#> 2170 240 125 85 sample_9_st_t 0.113      exonic      SRSF2
#> 2176 79 4 33 sample_9_st_t 0.142      exonic      JAK3
#>      ExonicFunc.refGene      cosmic70
#> 347 nonsynonymous_SNV      .
#> 391 synonymous_SNV      .
#> 2170 synonymous_SNV ID\|x3dCOSM4130674\|x3bOCCURENCE\|x3d1(thyroid)
#> 2176 synonymous_SNV      .
#>      non_cancer_AF_popmax Alt_dpGAP_PopFreq      v_name
#> 347      0.2566      . chr5:150068242:GAGC:G
#> 391      .      . chrX:48792153:G:C
#> 2170      1      . chr7:102255385:C:CAA
#> 2176      .      . chr7:102257898:C:CT
#>      loci      mut_sample
#> 347 chr5:150068242 chr5:150068242:GAGC:G/sample_10_st_g
#> 391 chrX:48792153 chrX:48792153:G:C/sample_10_st_g
#> 2170 chr7:102255385 chr7:102255385:C:CAA/sample_9_st_g
#> 2176 chr7:102257898 chr7:102257898:C:CT/sample_9_st_g

```

Basic Usage of *qcCHIP*

In this section, we use *qcCHIP* to test the results of select CHIP candidate with different setting of VAF, DP, or population. The resulting figures and comparison summary file will help user decide the optimal VAF, DP, or population metric for their dataset.

Run qcCHIP with change of minimum VAF

This section demonstrates the usage of *qcCHIP* when use different setting of minimum VAF.

```

# input file
input_path<- system.file("extdata","demo_input.txt",package="qcCHIP")
in_f <- read.table(input_path,sep="\t",header=T)

# create test directory
out_dir <- paste0(getwd(),"/vaf_test")
vaf_permut <- qcCHIP(in_f,out_path = out_dir
                     ,metric_min = 0,
                     metric_step = 0.02,
                     metric_max = 0.1,

```



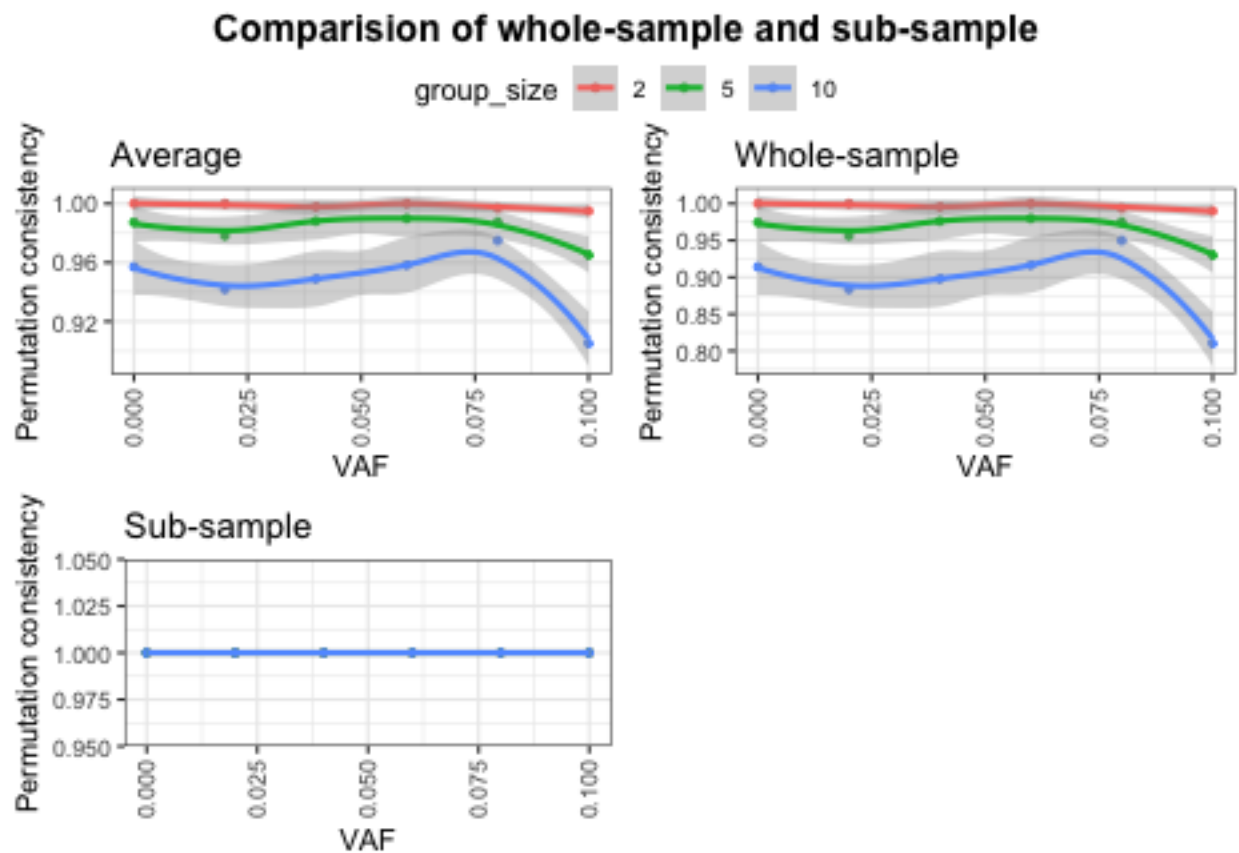
```

core=1,
show_info = F)

# example of comparision summary output
head(vaf_permut$summary_df)
#>   metric_name metric_setting group_size permut_index var_n_whole var_n_sub
#> 1      VAF      0      2      1      221      221
#> 2      VAF      0      2      2      221      221
#> 3      VAF      0      2      3      221      221
#> 4      VAF      0      2      4      221      221
#> 5      VAF      0      2      5      221      221
#> 6      VAF      0      2      6      221      221
#>   union_n common_n whole_only sub_only common_whole common_sub
#> 1      221      221      0      0      1      1
#> 2      221      221      0      0      1      1
#> 3      221      221      0      0      1      1
#> 4      221      221      0      0      1      1
#> 5      221      221      0      0      1      1
#> 6      221      221      0      0      1      1

# permutation consistency plot
vaf_permut$figs

```



Run qcCHIP with change of minimum DP

This section demonstrates the usage of *qcCHIP* when use different setting of minimum DP.

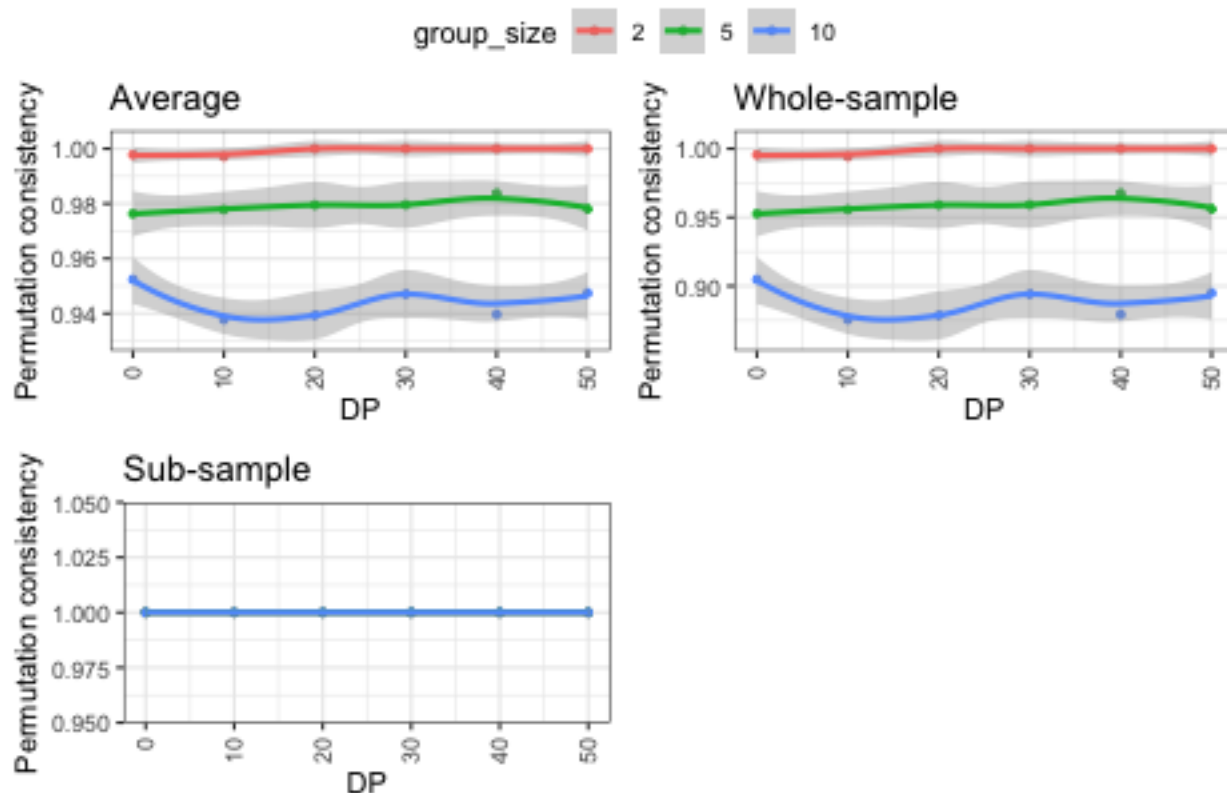
```
# input file
input_path<- system.file("extdata","demo_input.txt",package="qcCHIP")
in_f <- read.table(input_path,sep="\t",header=T)

# create test directory
out_dir <- paste0(getwd(),"/DP_test")
DP_permut <- qcCHIP(in_f,out_path = out_dir,permut_metrics = "DP",
                    metric_min = 0,
                    metric_step = 10,
                    metric_max = 50,
                    core=1,
                    show_info = F)

# example of comparision summary output
head(DP_permut$summary_df)
#>   metric_name metric_setting group_size permut_index var_n_whole var_n_sub
#> 1          DP              0           2           1         148        148
#> 2          DP              0           2           2         148        148
#> 3          DP              0           2           3         148        148
#> 4          DP              0           2           4         148        148
#> 5          DP              0           2           5         148        148
#> 6          DP              0           2           6         148        148
#>   union_n common_n whole_only sub_only common_whole common_sub
#> 1     148     148         0         0           1           1
#> 2     148     148         0         0           1           1
#> 3     148     148         0         0           1           1
#> 4     148     148         0         0           1           1
#> 5     148     148         0         0           1           1
#> 6     148     148         0         0           1           1

# permutation consistency plot
DP_permut$figs
```

Comparison of whole-sample and sub-sample



Run qcCHIP with change of maximum population percentage

This section demonstrates the usage of *qcCHIP* when use different setting of maximum population percentage.

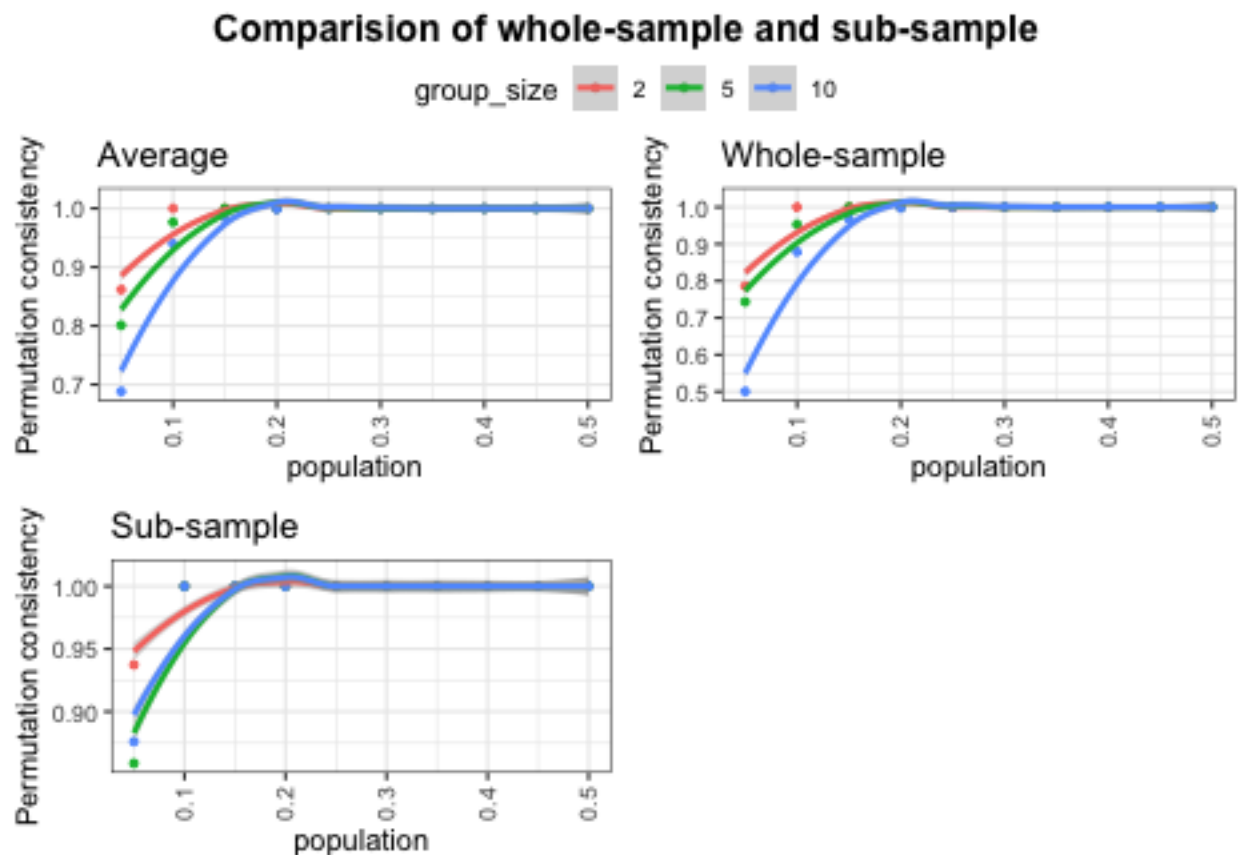
```
# input file
input_path<- system.file("extdata","demo_input.txt",package="qcCHIP")
in_f <- read.table(input_path,sep="\t",header=T)

# create test directory
out_dir <- paste0(getwd(),"/population_test")
pop_permut <- qcCHIP(in_f,out_path = out_dir,permut_metrics = "population",
  metric_min = 0.05,
  metric_step = 0.05,
  metric_max = 0.5,
  core=1,
  show_info = F)

# example of comparison summary output
head(pop_permut$summary_df)
#>  metric_name metric_setting group_size permut_index var_n_whole var_n_sub
#> 1  population          0.05           2           1         103         83
#> 2  population          0.05           2           2         103         92
#> 3  population          0.05           2           3         103         86
#> 4  population          0.05           2           4         103         76
```

```
#> 5 population 0.05 2 5 103 81
#> 6 population 0.05 2 6 103 83
#> union_n common_n whole_only sub_only common_whole common_sub
#> 1 103 83 20 0 0.806 1.000
#> 2 108 87 16 5 0.845 0.946
#> 3 113 76 27 10 0.738 0.884
#> 4 103 76 27 0 0.738 1.000
#> 5 103 81 22 0 0.786 1.000
#> 6 113 73 30 10 0.709 0.880

# permutation consistency plot
pop_permut$figs
```



Basic Usage of *CHIPfilter*

In this section, we use *CHIPfilter* to get the result of select CHIP candidate based on variety of selection matrices (detailed in the man page of *CHIPfilter*). The output will be a subset of input file which pass the selection. Users can directly use this function without running *qcCHIP*. Some features of *CHIPfilter* are described below.

```
# input file
input_path<- system.file("extdata","demo_input.txt",package="qcCHIP")
in_f <- read.table(input_path,sep="\t",header=T)
```

```

# blacklist region to exclude
bf_path<- system.file("extdata","demo_blacklist.bed",package="qcCHIP")

bl_f <- read.table(bf_path,sep = "\t",header=F)

# run default setting
out_1 <- CHIPfilter(in_f)
#> [1] "Perform population metrics"
#> [1] "Perform technique metrics"
#> [1] "No paired tumor sample, skip"
#> [1] "Perform functional metrics"
#> [1] "Perform not nonsunonymous metrics"
#> [1] "Perform gnomad metrics only"
#> [1] "No blacklist region bed file find, skip"

# change different metrics
out_2 <- CHIPfilter(in_f,max_percent=0.02,DP_min = 40,VAF_min=0.002,info=F)

# with paired tumor sample
out_3 <- CHIPfilter(in_f,tumor_sample = T,tumor_VAF_min = 0.02,info=F)

# with gnomad or dpGAP reference file
out_4 <- CHIPfilter(in_f,gnomad = F,dpGAP = F,info=F)

# with blacklist region
out_5 <- CHIPfilter(in_f,blacklist_f = bl_f,info=F)

# check the number of CHIP
length(unique(out_1$mut_sample))
#> [1] 148
length(unique(out_2$mut_sample))
#> [1] 71
length(unique(out_3$mut_sample))
#> [1] 103
length(unique(out_4$mut_sample))
#> [1] 148
length(unique(out_5$mut_sample))
#> [1] 145

```