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

2025-04-03

Contents

Introduction **1**

Install package **1**

Getting Started **2**

Preparing Input files **2**

Basic Usage of *qcCHIP* **4**

Basic Usage of *CHIPfilter* **8**

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 robust bioinformatics data filtering. We report an R package qcCHIP, a bioinformatics pipeline to guide and call CHs 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/RtmpTsHSMo/remotes1
#>   -  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 posation of variant.
3. End: end posation 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 synoymous 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.

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

# name of each variables
colnames(in_f)
```

```
#>  [1] "Chr"                  "Start"                "End"
#>  [4] "Ref"                  "Alt"                  "TLOD"
#>  [7] "SOR"                  "AD_alt"               "AF"
#> [10] "DP"                   "SAF"                  "SAR"
#> [13] "tumor_AF"             "SampleID"             "Func.refGene"
#> [16] "Gene.refGene"         "GeneDetail.refGene"   "ExonicFunc.refGene"
#> [19] "AAChange.refGene"     "cosmic70"             "non_cancer_AF_popmax"
#> [22] "ALT_dpGAp"            "Ref_dpGAP_PopFreq"    "Alt_dpGAP_PopFreq"

# value format of each variables
head(in_f)
#>     Chr     Start       End  Ref Alt   TLOD   SOR AD_alt    AF  DP SAF SAR
#> 1 chr16   3728301   3728304 CGCT   C  17.34 0.768      9 0.021 463   5   4
#> 2 chr17  31169974  31169974    C   A    3.3 4.975     11 0.051 142   0  11
#> 3  chrX  15820243  15820243    C   T 228.43 1.483     72 0.984  72  47  25
#> 4  chr7 140734494 140734494    T  TA    7.8 0.919     14 0.051 227   6   8
#> 5 chr21  43093000  43093000    G  GT   6.93 0.356      8 0.085  95  16   0
#> 6  chr7 102257421 102257422   GA   G   5.28 0.899      7 0.059 121   9   7
#>   tumor_AF   SampleID Func.refGene  Gene.refGene
#> 1    0.018 sample_155       exonic        CREBBP
#> 2    0.025 sample_155       exonic           NF1
#> 3    0.996 sample_155       exonic         ZRSR2
#> 4    0.000 sample_155         UTR3          BRAF
#> 5    0.141 sample_155         UTR3 U2AF1;U2AF1L5
#> 6    0.000 sample_155         UTR3          CUX1
#>
#> 1
#> 2
#> 3
#> 4
#> 5 NM_006758:c.*102C>AC;NM_001025203:c.*102C>AC;NM_001025204:c.*102C>AC;NM_001320650:c.*102C>AC;NM_00
#> 6
#>          ExonicFunc.refGene
#> 1 nonframeshift substitution
#> 2          nonsynonymous SNV
#> 3             synonymous SNV
#> 4                          .
#> 5                          .
#> 6                          .
#>                                                                                                                A
#> 1    CREBBP:NM_001079846:exon30:c.6629_6632delinsG:p.Q2210del,CREBBP:NM_004380:exon31:c.6743_6746del
#> 2 NF1:NM_000267:exon5:c.C563A:p.A188E,NF1:NM_001042492:exon5:c.C563A:p.A188E,NF1:NM_001128147:exon5:
#> 3                                                                                                ZRSR2:NM_005089:exon10:
#> 4
#> 5
#> 6
#>   cosmic70 non_cancer_AF_popmax ALT_dpGAp Ref_dpGAP_PopFreq Alt_dpGAP_PopFreq
#> 1        .              2.03E-05         .                 .                 .
#> 2        .                     .         .                 .                 .
#> 3        1                0.6222         .                 .                 .
#> 4        .                     .         .                 .                 .
#> 5        .                     .         .                 .                 .
#> 6        .                     .         .                 .                 .
```

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 comparision summary file will help user diceide the optimal VAF, DP, or population metric for their dataset.

Run qcCHIP with change of minimum VAF

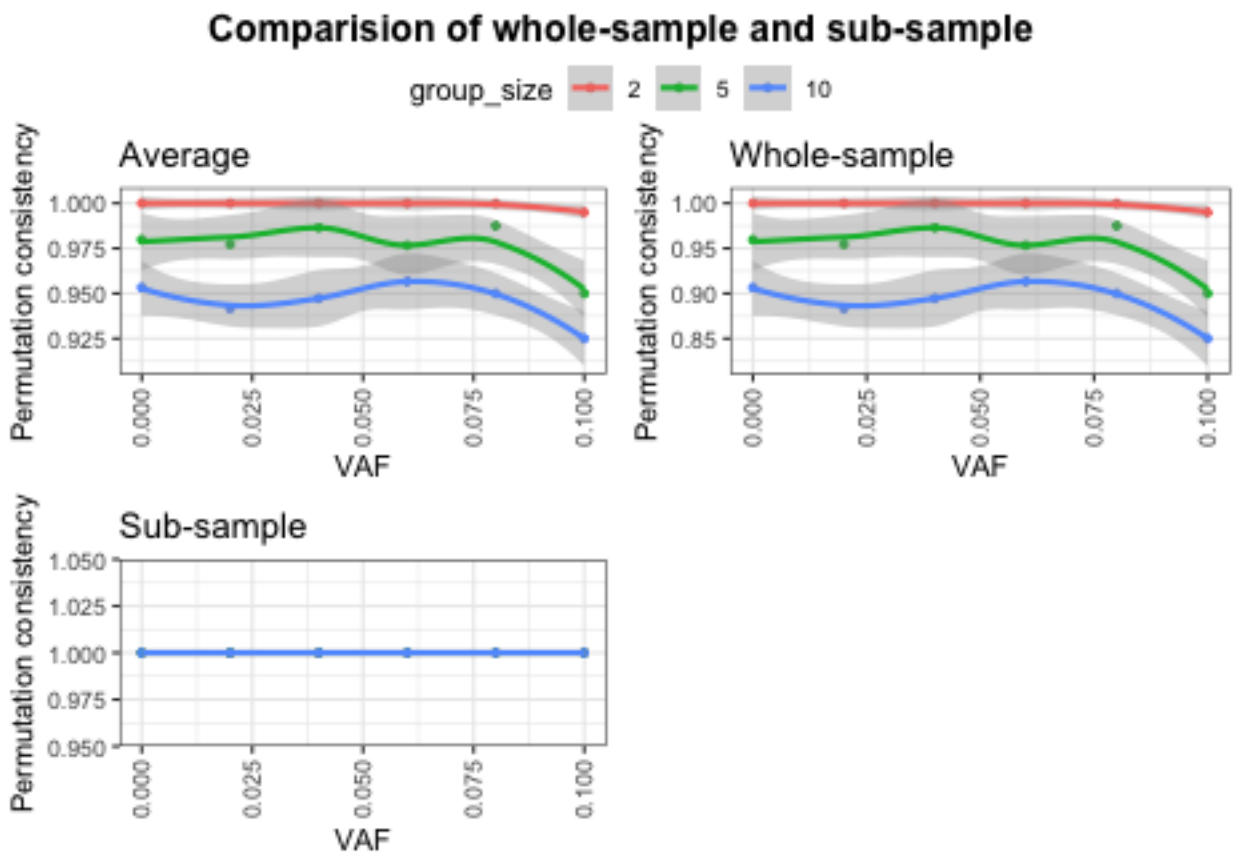
This section demonstrates the usage of *qcCHIP* when use different setting of minimum VAF.

```
# 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.

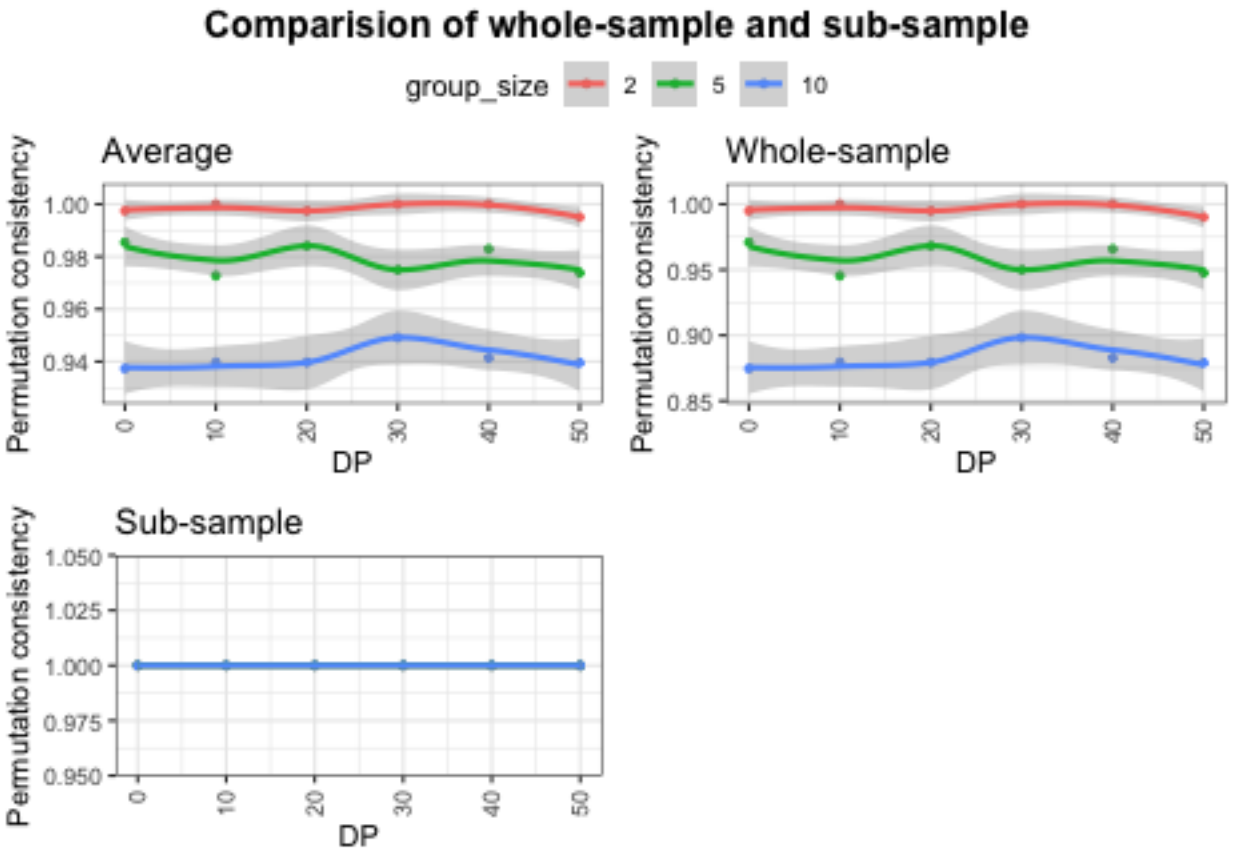
```
# input file
input_path<- system.file("extdata","demo_input.txt",package="qcCHIP")
in_f <- read.table(input_path,sep="\t",header=T)

# 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
```



Run qcCHIP with change of maximum population percentage

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

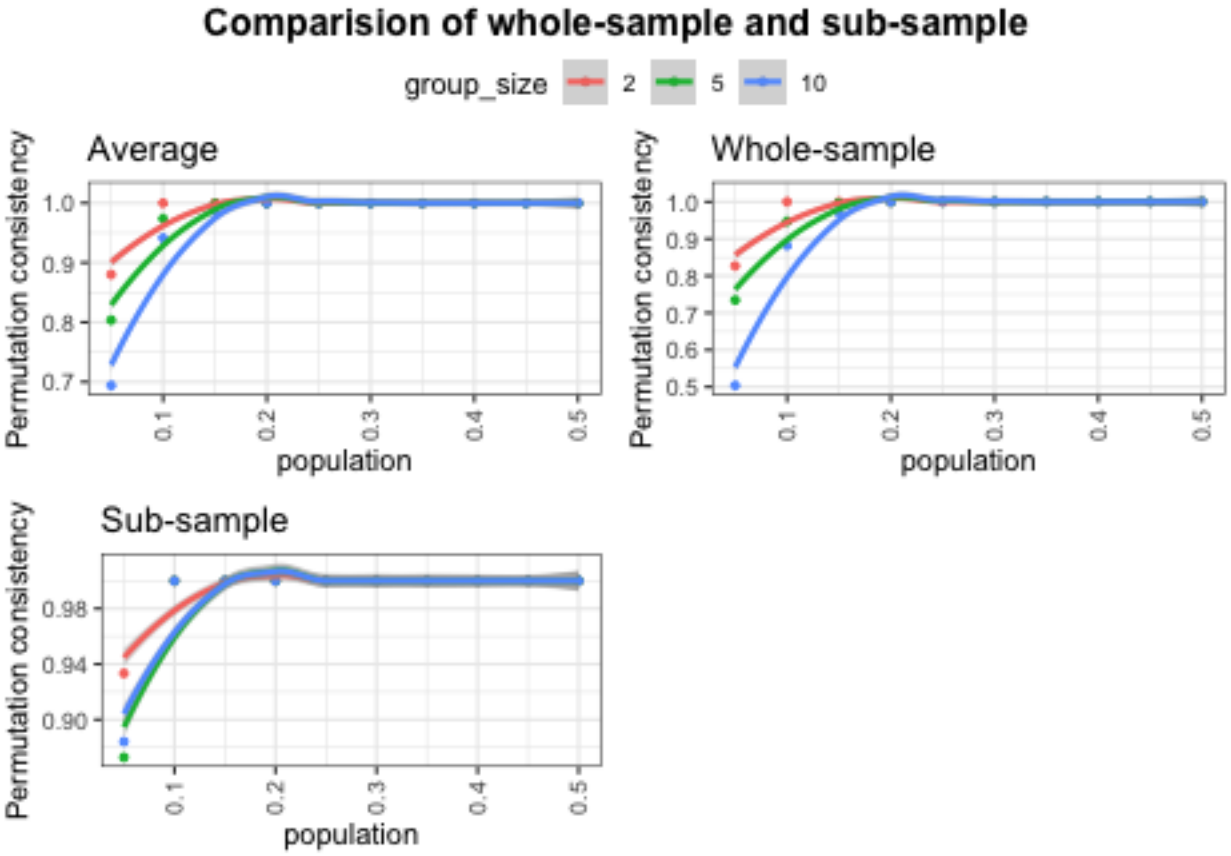
```
# input file
input_path<- system.file("extdata","demo_input.txt",package="qcCHIP")
in_f <- read.table(input_path,sep="\t",header=T)

# 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 comparision 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       100
#> 2  population           0.05          2            2         103        85
#> 3  population           0.05          2            3         103        87
#> 4  population           0.05          2            4         103        81
#> 5  population           0.05          2            5         103        93
#> 6  population           0.05          2            6         103        80
#>   union_n common_n whole_only sub_only common_whole common_sub
#> 1     113       90         13       10        0.874      0.900
#> 2     112       76         27        9        0.738      0.894
#> 3     103       87         16        0        0.845      1.000
#> 4     103       81         22        0        0.786      1.000
#> 5     107       89         14        4        0.864      0.957
#> 6     115       68         35       12        0.660      0.850

# 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 matrics (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 runing *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
```