Introduction of package 'MitoMutCall'

This R package is to calculate the substitution and heteroplasmy in the mitochondrial genome. First of all, BAM files are obtained from original fastq files by BWA program, through competitive alignment and mitochondrial circle processing; Secondly, Mpileup files are obtained from samtools program. Based on mpileup files, the minor allele (with lower allele frequency) will be reported as heteroplasmy if it passes the binomial test with a preset threshold. If the major allele (with higher allele frequency) is different to the allele in the mitochondrial reference genome (eg. revised Cambridge Reference Sequence, rCRS), it will be reported as substitution.

1. Installation and configuration

This package is based on Linux and R (>= 3.4.1). BWA and samtools are required to be pre-installed. Dependent R packages of "Biostrings", "data.table", "parallel", "stringr", "plyr", "reshape2" are also required.Only paired-end Sanger format FASTQ (Phred+33) files is supported.

Extract and enter the folder named 'MitoMutCall'; run the installation and configuration scripts

```
unzip MitoMutCall.zip
cd MitoMutCall
Rscript Installation_configuration.R
```

##Please ensure network connection. If the user is superuser, the R package will be installed to the ##default path; If the user is a regular user, the R package will be installed to ##"~/R/x86_64-pc-linux-gnu-library/" by default. If the dependent R packages have not been ##pre-installed, the program will download and install these packages by itself, as well as BWA ##and samtools in the configuration path. If some errors occur, it might be caused by no ##permissions to read and write the default folders. At this time, the users can enter to Rgui, and ##first sets the target path and then source(). Note: It can be set to any other path with read and ##write permissions.

```
.libPaths(paste("~/R/x86_64-pc-linux-gnu-library/",version$major,".",strsplit(version$minor,split=
".",fixed=TRUE)[[1]][1],sep=""))
source("Installation_configuration.R")
```

##You can download the dependent R packages for localized installation according to "list.txt".

```
R CMD INSTALL *(the relevant R packages)
```

##If you already have installed BWA or samtools, you can install or update the configuration by ##using the following functions. Set "bwa_path","samtools_path" with absolute path. Please note ##that bwa and samtools cannot be renamed.For example "/home/tools/bwa0.17.1/bwa"and ##"/home/tools/samtools1.5/samtools" will work.

```
Rscript Installation_configuration.R bwa_path samtools_path
```

2. Main functions

#call_bam() executes the fastq → bam → mpileup process through competitive alignment and #mitochondrial circle processing

call_bam(R1, R2, ref, ref_sequence_label ="MT", nThreads=20, competitive_alignment = TRUE, bam_index = FALSE, circle = TRUE, fq_sequence_length = 500, remove_dup=FALSE, mapQ=0, baseQ=30, bwa=paste(installed.packages()["MitoMutCall",c('LibPath')],"MitoMutCall/bwa", sep= "/"), samtools=paste(installed.packages()["MitoMutCall",c('LibPath')],"MitoMutCall/samtools", sep="/"))

#heteroplasmy() performs the pileup analysis from mpileup files to output substitution and #heteroplasmy files.

heteroplasmy(inpileup, rate=0.01, count=3, nThreads=20, name=NA, disputed_remove=TRUE, disputed_sites = c(302:317, 513:527, 566:574, 16181:16195, 3106, 3107, 16519))

Parameter definition:
R1： R1 mate fastq file for pair-ended reads (you can input multiple R1 fastq files for batch processing, but need to be matched to R2 accordingly).

R2： R2 mate fastq file for pair-ended reads (you can input multiple R1 fastq files for batch processing, but need to be matched to R1 accordingly).

ref: reference genome address (if "competitive_alignment" is TRUE, the reference genome contains genome reference sequences and mitochondrial reference sequence, and the mitochondrial sequence should be labeled in "ref_sequence_label"; If "competitive_alignment" is FALSE, the only sequence needs to provide as the mitochondrial reference genome with the "ref_sequence_label" can be omitted).

ref_sequence_label: mitochondrial reference sequence label, by default "MT".

nThreads: the number of threads, by default 20. When the number of threads is set to be greater than the limit number of threads in the server, the software automatically adjusts the number of threads to 90% of the limit of the server. When the number of threads is set to less than 2, the number of thread is automatically set to 1.

competitive_alignment: whether to conduct competitive alignment, by default TRUE. FALSE is not performed.

bam_index: whether there is a corresponding index file of BWA in the reference genome directory. If there is an index file in the directory, please set this parameter to TRUE to avoid re-index. If not, the software will copy the reference genome into the working directory and index it, by default FALSE.

circle: whether to perform circle processing. It will copy the starting part of the mitochondrial reference genome according to the base number set by "fq_sequence_length", and paste it to the end of the mitochondrial reference genome, by default TRUE. FALSE is not performed.

fq_sequence_length: when "circle =TRUE", this parameter is set as the base length of the reference genome of the mitochondria in circle processing.

remove_dup： whether to remove redundancy sequence. If TRUE, it will remove redundancy; If not, it will be not performed by default FALSE.

mapQ：the threshold of mapping quality score. For a particular short sequence read, consider its best alignment in the genome. For this alignment, calculate the sum of base quality scores at mismatched bases and define a quantity SUM_BASE_Q(best). I will also consider all other possible alignments for the read. For the alignment i, define SUM_BASE_Q(i) as the sum of base quality scores at mismatched bases for that alignment.
mapQ=-log10 {1.0 – [10^(- SUM_BASE_Q(best))]/[∑i10^(- SUM_BASE_Q(i))]}

baseQ: the threshold of Base Quality Score.

bwa : the path to BWA software; by default it is built-in in the package.

samtools: the path to samtools software; by default it is built-in in the package.

inpileup: mpileup input file.

rate: frequency threshold of heteroplasmy, by default 0.01.

count: minimum threshold of number of minor alleles at R1 and R2 end, respectively,by default 3.

name: in general, it is not required to set the prefix name of the output substitution and heteroplasmy file, and the default is mpileup file name prefix. If the prefix name needs to be changed,itwill be set as a dynamic variable to avoid the previous result file being overwritten by the later one in batch processing.

disputed_remove: whether to remove controversial sites, TRUE is to remove controversial sites, and FALSE is not to remove. And the result files will be saved with name of "*_disputed_remove_hmarker.txt", by default TRUE.

disputed_sites: if and only if disputed_remove = TRUE, remove the default controversial site " c(302:317,513:527,566:574,16181:16195,3106,3107,16519)". Users can change the removed sites.

3.Details of competitive alignment

Alignment of mitochondrial genomes（MT）, Alignment of nuclear genomes（nuclear）， and no alignment of nuclear genomes（*）

Fastq files are competitively aligned. Mitochondria and unaligned reads were retained.And we remove theperfect alignments to the nuclear genomes.

The details are as follows:

Different alignment	R1	R2	Trade-off situation
All alignments in MT	MT	MT	retained
Perfect alignments to the nuclear genomes	nuclear	nuclear	removed
Alignments to the different positions of the nuclear genomes	nuclear	nuclear	retained
No alignments	*	*	retained
The read aligned to the mitochondria, the mate aligned to the nuclear genomes	MT	nuclear	retained
	nuclear	MT	
The read aligned to the mitochondria, no alignment for the mate	MT	*	retained
	*	MT	
The read aligned to the nuclear genomes, no alignment for the mate	nuclear	*	retained
	*	nuclear	

4.Output results

After the program is executed successfully, a folder named "heteroplasmy_result" will be created in the working directory. The final output resultsare saved here.

*_smarker.txt：Results of all substitution mutations

*_hmarker.txt：Results of all heteroplasmy mutations that meet the requirements

*_disputed_remove_hmarker.txt：Results of heteroplasmy mutations after removingsuspected part

When there is no substitution mutation or heteroplasmymutation in the detected sample, it will be displayed on the screen during the program running:：

"There is no substitution sites in (sample name)" or "There is no heteroplasmy sites in (sample name)"

Parameter description of result tables	
Parameter	Description
loc	Locus of substitution or heteroplasmy
ref	The corresponding base on the reference genome
UA	Count of A base aboutthe corresponding locus in plus strands
UT	Count of T base aboutthe corresponding locus in plus strands
UC	Count of C base aboutthe corresponding locus in plus strands
UG	Count of G base aboutthe corresponding locus in plus strands
UAN	Proportion of A base aboutthe corresponding locus in plus strands
UTN	Proportion of T base aboutthe corresponding locus in plus strands
UCN	Proportion of C base aboutthe corresponding locus in plus strands
UGN	Proportion of G base aboutthe corresponding locus in plus strands
AssU	The major allele of the plus strand
Ucount1	Count of the major allele of the plus strand
Ucount2	Count of the minor allele of the plus strand
Urate2	Frequency of the minor allele of the plus strand
Ualt	The minor allele of the plus strand
LA	Count of A base aboutthe corresponding locus in minus strands
LT	Count of T base aboutthe corresponding locus in minus strands
LC	Count of C base aboutthe corresponding locus in minus strands
LG	Count of G base aboutthe corresponding locus in minus strands
LAN	Proportion of A base aboutthe corresponding locus in minus strands
LTN	Proportion of T base aboutthe corresponding locus in minus strands
LCN	Proportion of C base aboutthe corresponding locus in minus strands
LGN	Proportion of G base aboutthe corresponding locus in minus strands
AssL	the major allele of the minus strand
Lcount1	Count of the major allele of the minus strand
Lcount2	Count of the minor allele of the minus strand
Lrate2	Frequency of the minor allele of the minus strand
Lalt	The minor allele of the minus strand
Smarker	Whether substitution mutation occurs at this site, 1 indicates substitution mutation and 0 indicates no substitution mutation
pvalue	The pvalue of binomial test
heteroplasmy	Heteroplasmyestimates by binomial test
U_heteroplasmy	Heteroplasmyestimates of the plus strand by binomial test
L_heteroplasmy	Heteroplasmyestimates of the minus strand by binomial test

5.Examples：

```
#If the package is not installed into the default path, Please use .LibPaths () to load the path before
#launching the package.

.libPaths(paste("~/R/x86_64-pc-linux-gnu-library/",version$major,".",strsplit(version$minor,split=
".",fixed=TRUE)[[1]][1],sep=""))
library("MitoMutCall")

# Example.1 Processes 1 pair fastq files in the working directory
##It is assumed that there are 1 pair of files "R1.fq", "R2.fq" and reference genome "MT.fa" with
##an uniquemitochondrial sequence in theworkingdirectory.Set competitive_alignment= FALSE,
##circle = FALSE, remove_dup = TRUE. Therefore, the following code shows that the process
##include removing duplication, but no competitive alignment and mitochondrial circle
##processing.

call_bam(R1="R1.fq", R2="R2.fq", ref="MT.fa", competitive_alignment=FALSE, circle=FALSE,
remove_dup=TRUE)

## Disputed_remove = FALSE can be set, then the result of *_disputed_remove_hmarker.txt does
## not output.
## If rate=0 and count=0, all suspected heteroplasmy mutations will be output without screening.

for (inpileup in
      grep(".mpile.file$",list.files(),value=TRUE)){ heteroplasmy(inpileup,rate=0,count=0,
      disputed_remove=FALSE)
}

# Example.2 Processes multiple pairs of fastq files in the working directory
## It is assumed that there are 2 pair of files 1.R1.fq, 1.R2.fq, 2.R1.fq, 2.R2.fq and reference ##
##genome MT.fa in the working directory. Set competitive_alignment= FALSE, mapQ=20,
##baseQ=30. Therefore, the following output shows that the process donot include competitive
##alignment. The bases are selected according to the quality values set by mapQ and baseQ.

call_bam(R1=c("1.R1.fq","2.R2.fq"),R2=c("1.R2.fq","2.R2.fq"),mapQ=20,baseQ=30,ref="MT.fa",
competitive_alignment = FALSE)
for (inpileup in
      grep(".mpile.file$",list.files(),value=TRUE)){ heteroplas
      my(inpileup)
}

#    Example.3 Batch processing of fastq/fasta files in the non-working directory
## Set the path of fastq

Path0="/home/fastq"
```

```
##Set the reference genome path. If there is bwa index file inthe reference genome path
##bam_index=TRUE; otherwise, please set to FALSE. If competitive alignment is preferred,
##ref_sequence_labelshould be labelled.as like ref_sequence_label="MT".

Ref_path="/home/fasta/Homo_sapien.fa"

##If you want to use another version of BWA or samtools, but do not want to
##reconfigure with the R package, you can invoke the software by specifying ## the path of BWA
##or samtools

bam_path="/home/bwa"
samtools_path="/home/samtools"

## *1. Fq or *2. Fq are assumed to be matched fastq files at the R1 end or R2 end, ensuring that
##R1 files matchedto R2 files accordingly. The following code runs the program with 20 threads
##by default.

call_bam(R1=list.files(path=Path0, "*1.fq$", full.names=TRUE), R2=list.files(path=Path0,
"*2.fq$", full.names=TRUE), nThreads=20, ref_sequence_label="MT", mapQ=0, baseQ=30,
ref=Ref_path, bam_index=FALSE, bwa= bam_path, samtools= samtools_path)
for (inpileup in
    grep(".mpile.file$",list.files(),value=TRUE)){ heteropla
    smy(inpileup,nThreads=20,rate=0.01)
}
```