Introduction of package ‘MitoMutCall’

The function in this R package is to calculate the mutations of substitution and heteroplasmy in mitochondria. 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, minor allele frequency is used as the mitochondrial heteroplasmy through binomial test, and if the major allelic bases are different of reference, those bases are substitution mutations.

1. Installation and configuration

This package is based on Linux Linux and R (>= 3.0.1). The softwares of BWA and samtools are required to be installed; And the relevant R packages of "Biostrings", "data.table", "parallel", "stringr", "plyr", "reshape2" are needed. The FASTQ files supported by this software are only limited to the paired-end reads with the standard Phred+33 quality score system.

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 by itself; 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 relevant R packages have not been
## installed before, the program will download and install these packages by itself. BWA and
## samtools software in the configuration path are installed by default. If it occurs some errors of
## R package installation, it may be owing to users haven’t the permissions to read and write
## 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 relevant R packages for localized installation according to "list.txt".

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

## If you already have installed BWA or samtools in system, you can install or update the
## configuration by using the following functions. Set "bwa_path", "samtools_path" to
## absolute path, bwa and samtools cannot be renamed. Such as "/home/tools/bwa0.17.1/bwa" and
## "/home/tools/samtools1.5/samtools" meet the requirements. There are no matters that the input
## order of "bwa_path" and "samtools_path". Users can enter one or two paths

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

2. Main functions

# call_bam() mainly realizes the transformation from fastq → bam → mpileup through ## competitive alignment and mitochondrial circle processing

call_bam(R1, R2, ref, ref_sequence_label = "MT", nThreads = 20, competitive_alignment= TRUE, bam_index = TRUE, circle = TRUE, remove_dup = FALSE, fq_sequence_length = 500, 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() mainly realizes the transformation from mpileup files→pileup analysis files # →the result files of substitution and heteroplasmy

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 end fastq file address for pair-ended reads (you can input vectors for batch processing, but need to correspond to R2 one by one).

R2：R2 end fastq file address for pair-ended reads (you can input vectors for batch processing, but need to correspond to R1 one by one).

ref: reference genome address (if "competitive_alignment" is TRUE, the reference genome contains multiple reference sequences, and the mitochondrial sequence label shall be filled in at "ref_sequence_label"; If "competitive_alignment" is FALSE, the reference genome can be directly set as the unique mitochondrial reference genome with only one sequence of mitochondrion. ref_sequence_label can be omitted).

ref_sequence_label: mitochondrial sequence label in entire reference genome, default to "MT".

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

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

bam_index: determine 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 bam_index is set to FALSE, the software will copy the reference genome into the working directory and index it, default to TRUE.

circle: determine whether circle processing; When TRUE, 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, default to TRUE.

remove_dup： determine whether to remove redundancy sequence. TRUE means to remove redundancy; FALSE does not execute, default to FALSE.

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

mapQ：Set 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). 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))]/[∑i 10^(- SUM_BASE_Q(i))]}

baseQ: Set the threshold of Base Quality Score.

bwa: the path to BWA software; Default is built-in to the package.

samtools: the path to samtools software; Default is built-in to the package.

inpileup: mpileup input file.

rate: frequency threshold of heteroplasmy. The default value is 0.01. When the rate is set to 0, any site with heteroplasmy will be output.

count: minimum threshold of minor alleles at R1 and R2 end, respectively. The default is 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, it will be set as a dynamic variable to avoid the previous result file being overwritten by the later one with the same name when batch processing.

disputed_remove: To remove controversial sites, TRUE is to remove controversial sites, and FALSE is not to remove. If "disputed_remove = TRUE", the result files will be saved with name of "*_disputed_remove_hmarker.txt". The default value is TRUE.

disputed_sites: If and only if disputed_remove = TRUE, remove the default controversial site "c (302:317513-527566:574161, 81:16195310, 6310, 7165 (19) ". 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 the perfect 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
One end aligned to the mitochondria, the other to the nuclear genomes	MT	nuclear	retained
	nuclear	MT	
One end aligned to the mitochondria, the other not aligned	MT	*	retained
	*	MT	
One end aligned to the nuclear genomes, the other not aligned	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 results are 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 removing suspected part
When there is no substitution mutation or heteroplasmy mutation 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 about the corresponding locus in plus strands
UT	Count of T base about the corresponding locus in plus strands
UC	Count of C base about the corresponding locus in plus strands
UG	Count of G base about the corresponding locus in plus strands
UAN	Proportion of A base about the corresponding locus in plus strands
UTN	Proportion of T base about the corresponding locus in plus strands
UCN	Proportion of C base about the corresponding locus in plus strands
UGN	Proportion of G base about the 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 about the corresponding locus in minus strands
LT	Count of T base about the corresponding locus in minus strands
LC	Count of C base about the corresponding locus in minus strands
LG	Count of G base about the corresponding locus in minus strands
LAN	Proportion of A base about the corresponding locus in minus strands
LTN	Proportion of T base about the corresponding locus in minus strands
LCN	Proportion of C base about the corresponding locus in minus strands
LGN	Proportion of G base about the 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	Heteroplasmy estimates by binomial test
U_heteroplasmy	Heteroplasmy estimates of the plus strand by binomial test
L_heteroplasmy	Heteroplasmy estimates of the minus strand by binomial test

5.Example：

#If the installation package is not in the default path, Please use .LibPaths () to load the path before reading 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
## a unique sequence of mitochondrion in the working directory.
## 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 (and the reference genome has only one mitochondria
## reference genome).
## Set competitive_alignment= FALSE, mapQ=20, baseQ=30. Therefore, the following output
## shows that the process do not 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)){
	heteroplasmy(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 under the reference genome path
## bam_index=TRUE; otherwise, should be set to FALSE. And if competitive alignment is
## required, ref_sequence_label should be set to accurate labeling in the reference genome. Just
## like ref_sequence_label="MT".

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

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

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

## *1. Fq or *2. Fq are assumed to be the uniform format of fastq files at the R1 end or R2 end,
## ensuring that R1 files correspond to R2 files one by one.
## Then the following code runs the program with 20 threads as the default parameter.

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)){
    heteroplasmy(inpileup,nThreads=20,rate=0.01)
}
```