Get_MR1.0

1. 写在前面：

1.1 项目地址

github：HaobinZhou/Get_MR: A package for running MR In batches and in parallel quickly (github.com)

如果觉得好用，可以点一下github项目上的小星星吗，这是我们继续开源的最大动力，谢谢！

1.2 R包使用方法：

R包以R脚本的形式提供，打开R包，全选运行，即得到所有function

1. 进入githubHaobinZhou/Get_MR: A package for running MR In batches and in parallel quickly (github.com)，下载代码zip
2. ```
   source("./Get_MR1.0.r") ## 填文件所在地址

   ## 或者直接打开R文件，全选代码运行也可以！
   ```

1.3 常见问题：

1. **本地clump，1000G处理好的MAF文件，MRlap依赖文件如何获取**： GetScience公众号可免费获取已处理好文件，回复"依赖文件"即得链接。源文件请查看本文相应function介绍处
2. **输入clump文件路径后总是报错**:

   ```
   #尤其注意这个文件名的书写，因为他们是二进制文件，不需要写后缀！只需要选取对应的人种即可，比如欧洲人：
   LD_file="S:/GWAS数据/本地LD依赖文件/EUR"

   ## 这个问题我回答好多遍啦！
   ```

3. **第一次使用如何安装关联R包**：Get_MR/1.0 at main · HaobinZhou/Get_MR (github.com)
   1. 如果不需要使用`MungeSumstats`包（相关函数包括：`format_Mun`，`get_chr_pos`，`format_getmr`中`source="ukb_nosnp"`），则只需要运行Get_MR1.0dependence.R
   ```

2. 如果需要使用`MungeSumstats`包，则还需运行Install_Reference_Genome.r 这个包括了hg19和hg38的基因组参考文件，总大小达到了5G！**如果直接安装失败，在GetScience公众号回复"基因组参考"可得下载链接，并本地安装**（推荐）

4. **Bug反馈**：代码仅由两人编写，难免出现错误。欢迎提交bug到GetScience公众号后台！
5. **感谢所有Get_MR使用的R包作者**，是因为他们我们才得以轻松实现这么多复杂的功能。他们都是开源的，因此我们承诺Get_MR将**永久免费开源**。这意味着使用者可以随意地修改，分发代码，但前提是遵守：

   1.本代码不得用于任何商业或盈利目的

   2.未经代码作者的同意，本代码不得用于任何形式的销售或商业交易

   3.本代码可以在非商业性的科研、学术研究和个人使用的情况下免费使用

   4.在使用本代码并重新打包并向公众发放时，请引用我们的公众号原文

2. 帮助文档目录

进阶MR分析

LDSC_rg

用于计算两个数据框中SNP之间的遗传相关性（rg）。

用法

```
LDSC_rg(expo, outcome, an, sample_prev = NA, population_prev = NA,
        ld, wld, chr_filter = c(1:22), n_blocks = 200)
```

参数

- `expo`: 一个数据框，其中包含一个遗传暴露指标的多个SNP和它们与结果变量的rg。
- `outcome`: 一个数据框，其中包含一个结果变量的多个SNP和它们与遗传暴露指标的rg。
- `an`: 它是一个字符串,目前还没有作用（因为我们提供的依赖文件只有eur的，其他人种还没更新）
- `sample_prev`: 遗传暴露指标的样本流行病学先验患病率。默认为 `NA`。
- `population_prev`: 遗传暴露指标的人群流行病学先验患病率。默认为 `NA`。
- `ld`: 本地LD依赖文件
- `wld`: 本地weighted LD 依赖文件
- `chr_filter`: 一个整数向量，用于指定要使用的染色体。默认为包含1-22的整数向量。
- `n_blocks`: 用于计算加权LD矩阵的块数。默认为200。

返回值

一个具有以下元素的列表：

- `rg`: 两个数据框中SNP之间的遗传相关性（rg）。
- `pval`: `rg` 的双侧P值。
- `N_snps`: 参与计算rg的SNP数量。

示例

具体用法参照：mr_lap和LDSC_rg示例.r 可通过公众号GetScience回复示例获取文件

mr_lap

描述

mrlap是一种矫正样本重叠后的双样本MR方法。可用于怀疑有样本重叠的数据中。

R包官网：[n-mounier/MRlap: R package to perform two-sample Mendelian Randomisation (MR) analyses using (potentially) overlapping samples (github.com)]

语法

```
mr_lap(expo, outcome, ld, hm3, pval, r2, kb, MR_reverse = 1e-03, save_logfiles = F)
```

参数

- `expo`: 数据框，为TwoSampleMR包格式的数据
- `outcome`: 数据框，为TwoSampleMR包格式的数据
- `ld`: 数据框，本地LD文件路径
- `hm3`: 数据框，本地HapMap3文件路径
- `pval`: 数值，MR 工具变量阈值。
- `r2`: 数值，clump阈值
- `kb`: 数值，clump阈值
- `MR_reverse`: 数值，MR 的方向翻转阈值。
- `save_logfiles`: 逻辑值，是否保存日志文件。

值

- res: mrlap 结果。

用法

具体用法参照：mr_lap和LDSC_rg示例.r 可通过公众号GetScience回复示例获取文件

cause_getmr函数

描述

一键式执行cause。可批量化执行多暴露对一结局或一暴露对多结局

用法

```
## 不并行化运行
cause_getmr(expo, outcome, LD_file, r2 = 0.001, kb = 10000, pval = 1e-05)

## 并行化运行
cl<-makeCluster(2) ## 填你想要的并行化的核数，核数越多，需要的运行内存越大
cause_getmr(expo, outcome, LD_file, r2 = 0.001, kb = 10000, pval = 1e-05,cl=cl)
```

参数

- `expo`: TwoSampleMR的暴露格式的数据。
- `outcome`: TwoSampleMR的暴露格式的数据。

- 注意！expo和outcome，可以是data.frame的形式，也可以是一个list（如list[[1:n]]里都包含数据的data.frame)。但不能outcome和expo同时都是list。当expo或outcome，其中一个为list的情况下，是批量运行一对一的cause。比如我读取了10个暴露和1个结局，将10个暴露lapply读取进来就会是一个list。这时候 `cause_cyclemr` 自动运行每个暴露对结局的cause，也就是批量化执行.

```
## 比如我这里读取3个暴露文件和1和结局文件
id<-c('a.gz','b.gz','c.gz')
expo<-lapply(id,FUN=fread)
outcome<-fread("outcome.gz")
cl=makeCluster(4)## 内存不够的也可以不并行化运行
res<-cause_getmr(expo, outcome, LD_file, r2 = 0.001, kb = 10000, pval = 1e-05,cl=cl)
stopCluster(cl)
## 这样返回的结果就是3个暴露分别对一个结局的cause结果。
```

- `LD_file`: 包含LD信息的PLINK文件名。因为需要大批量地clump，在线clump很容易报错，因此我们采用本地clump。需要本地参考文件。下载地址： http://fileserve.mrcieu.ac.uk/ld/1kg.v3.tgz 。 或关注公众号GetScience直接获取。

```
#尤其注意这个文件名的书写，因为他们是二进制文件，不需要写后缀！只需要选取对应的人种即可，比如欧洲人：
LD_file="S:/GWAS数据/本地LD依赖文件/EUR"

## 这个问题我回答好多遍啦！
```

- `r2`: LD的R平方阈值。默认值为0.001。
- `kb`: LD的距离阈值（以kb为单位）。默认值为10000。
- `pval`: 用于LD clumping的p值阈值。默认值为1e-05。
- `cl`: 并行计算的cluster对象。默认值为NULL。在外部使用cl=makeCluster(n),n为你想并行化的核数。注意核数太多不要爆内存了。

值

`cause_cyclemr` 函数返回cause结果

RAPS_getmr

描述

`RAPS_getmr` 函数执行基于RAPS的MR并返回结果,并画图

用法

```
expo<-fread('a.gz')
outcome<-fread('b.gz')
expo<-format_data(...)
outcome<-format_data(...)  ## format_data是TwoSampleMR包的函数，格式化。
expo<-pblapply(expo,pval=1,kb=10000,r2=0.001,LD_file=LD_file,FUN=clean_expo) ##数据很大，建议本地clump，在线很容易报错
dat<-harmonise(expo,outcome)
res<-RAPS_getmr(dat, dir_figure)
```

参数

- `dat`: TwoSampleMR包 harmonise_data后输出的数据
- `dir_figure`: 保存结果图形的目录。

值

`RAPS_getmr`函数返回一个包含基于RAPS的MR结果的数据框。

mr_Presso

描述

执行MR-PRESSO

语法

```
mr_Presso(dat, num = 10000)
```

参数

- `dat`: 数据框，包含基因表达和疾病风险关联分析的数据。
- `num`: 整数，模拟数量。

值

- `mr_presso_res`: MR-PRESSO 结果。

用法

```
dat<-harmonise_data(exposure,outcome) ## TwoSampleMR包的harmonise_data函数输出的结果
mr_presso_res <- mr_Presso(dat, num = 10000)
```

mr_presso_pval函数

描述

提取 MR-PRESSO 结果中的主要结果

语法

```
mr_presso_pval(mr_presso_res)
```

参数

- `mr_presso_res`: MR-PRESSO 结果。

值

- mr_presso_main: MR-PRESSO 主要结果。

用法

```
mr_presso_main <- mr_presso_pval(mr_presso_res) ##mr_Presso输出的结果
```

mr_presso_snp函数

描述

根据 MR-PRESSO 分析结果，将离群值剔除，返回剔除离群值后的dat(我一般称为dat_aj， 也就是adjusted_data)， 可用于后续的IVW等分析。

语法

```
mr_presso_snp(mr_presso_res, mr_presso_main, dat, type = "list")
```

参数

- `mr_presso_res`: MR-PRESSO 结果。
- `mr_presso_main`: MR-PRESSO 主要结果。
- `dat`: 数据框或数据框列表，包含基因表达和疾病风险关联分析的数据。
- `type`: 字符串，输入数据类型。可选值为 "list" 或 "data"。如果是列表形式的（批量化运行后的结果），就是`list`，如果是普通数据框就是data

值

过滤后的数据框或数据框列表。

用法

```
dat<-harmonise_data(exposure,outcome) ## TwoSampleMR包的harmonise_data函数输出的结果
mr_presso_res <- mr_Presso(dat, num = 10000)
mr_presso_main <- mr_presso_pval(mr_presso_res)
data_aj <- mr_presso_snp(mr_presso_res, mr_presso_main, dat, type = "data")

## 用矫正的data可以用于后续的分析，例如重新计算mr
res_aj<-mr(data_aj)
```

快捷预处理及质控工具

format_Mun

介绍

运用MungeSumstats包标准化GWAS 摘要统计数据（包括hg19和hg38转换）。该函数可以将来自Finngen R8和其他来源的 GWAS 摘要统计数据文件清洗为标准的GWAS文件，并可将基因组位置从`ref_genome`转换到`convert_ref_genome`。

用法

```
format_Mun(file, source = "finn_r8", save_path = NULL, lift = F, ref_genome =
"hg38", convert_ref_genome = "hg19")
```

参数

- `file`：字符向量或数据框，表示要格式化的 GWAS 摘要统计数据文件或数据框。如果输入的是字符向量，则表示文件的路径。如果输入的是数据框，则表示要格式化的数据框。
- `source`：字符向量，表示输入文件的来源。默认为 `"finn_r8"`。
- `save_path`：字符向量，表示格式化文件要保存的路径。默认为 `NULL`。
- `lift`：逻辑值，表示是否将基因组位置从 `ref_genome` 转换到 `convert_ref_genome`。默认为 `F`。
- `ref_genome`：字符向量，表示 GWAS 摘要统计数据文件使用的参考基因组。默认为 `"hg38"`。
- `convert_ref_genome`：字符向量，表示要将基因组位置转换到的参考基因组。默认为 `"hg19"`。

例子

```
# 从文件中格式化数据
format_Mun("my_sumstats.txt", save_path = "~/formatted_sumstats", lift = F,
ref_genome = "hg38")

# 从数据框中格式化数据
my_sumstats_df <- read.csv("my_sumstats.csv")
format_Mun(my_sumstats_df, save_path = "~/formatted_sumstats", lift = F,
ref_genome = "hg38")

#格式化数据并升降版本
format_Mun(my_sumstats_df, save_path = "~/formatted_sumstats", lift = T,
ref_genome = "hg38", convert_ref_genome = "hg19") ## 从hg38转为hg19
```

返回值

该函数返回格式化的数据框并将其写入磁盘文件。`save_path`指定保存的位置

format_getmr

介绍

预设的快捷格式化 GWAS 摘要统计数据，这个函数用于将来自多个数据来源的 GWAS 摘要统计数据转换为 TwoSampleMR 包所需的格式。

用法

```
format_getmr(data, type = "exposure", source = "finn_r8")
```

参数

- `data`：数据框，表示要格式化的 GWAS 摘要统计数据。
- `type`：字符向量，表示数据类型，可以是 "exposure" 或 "outcome"。默认为 "exposure"。
- `source`：字符向量，表示数据来源。默认为 "finn_r8"。目前支持的来源有：
  - "finn_r8": Data download - FinnGen Documentation (gitbook.io)
  - "ukb_nosnp": 尼尔数据库（UKB），因为没有rsid，因此需要匹配（已一键完成）。www.nealelab.is/uk-biobank
  - "Mun"： 来自MungeSumstats包格式化后的数据
  - "covid"： COVID19-hg GWAS meta-analyses round 7 (covid19hg.org)
  - "outcome"： 已经格式化为TwoSampleMR包的“outcome”格式
  - "exposure"：已经格式化为TwoSampleMR包的“exposure”格式
  - "fast_ukb": fastGWA | Yang Lab (westlake.edu.cn)
  - "bac": 2021年肠菌原文数据 Large-scale association analyses identify host factors influencing human gut microbiome composition - PMC (nih.gov)

例子

```
my_data <- fread("my_data.gz")
format_getmr(my_data, type = "finn_r8", source = "Mun")
```

返回值

该函数返回格式化的数据框。

format_trait

介绍

这个函数用于格式化 GWAS 摘要统计数据中的表型信息，使其符合命名规范，易于保存为文件（例如批量保存计算R2和F值后的文件）。

主要是为了解决，在Windows系统下，保存文件的名称中不能包含特殊字符，例如`:`,`|`。

用法

```
format_trait(list, short = FALSE, short_num = "40")
```

参数

- `list`：列表，表示要格式化的 GWAS 摘要统计数据列表。
- `short`：逻辑值，表示是否要将表型名称缩短。默认为 FALSE。
- `short_num`：字符向量，表示缩短表型名称的长度。默认为 "40"。

例子

```
my_list <- list(data1, data2, data3)
format_trait(my_list, short = TRUE, short_num = "20")
```

返回值

该函数返回格式化后的 GWAS 摘要统计数据列表。

read_vcf_getmr

介绍

这个函数用于从 VCF 文件中读取摘要统计数据。并保存为压缩文件。默认是.gz为后缀的压缩文件。方便下次读取以及节省空间。

这是因为读取VCF文件将消耗大量电脑资源。我们建议批量读取VCF文件后储存为易于读取的压缩包形式。下次读取方便快捷。因此本函数不会直接返回数据框，而是保存为文件

用法

```
read_vcf_getmr(file_name, nThread = 8, type = ".gz")
```

参数

- `file_name`：字符向量，表示要读取的 VCF 文件名。
- `nThread`：整数，表示要使用的线程数。默认为 8。
- `type`：字符向量，表示输出文件类型。默认为 ".gz"。

例子

```
my_file <- "my_file.vcf"
read_vcf_getmr(my_file, nThread = 4, type = ".gz")
```

返回值

该函数没有返回值，而是将读取的数据写入文件。

read_easy

介绍

这个函数用于从文件中读取 GWAS 摘要统计数据。并返回经过P值筛选的文件。一般用于批量读取大量文件时。比如我要批量读取100个暴露数据，每个数据占用运行内存2G。如果100个，则200G，不是一般电脑可以承受。因此每次读取将直接筛选p值，压缩大小

用法

```
read_easy(file_name, pval = 5e-08)
```

参数

- `file_name`：字符向量，表示要读取的文件名。
- `pval`：数字，表示筛选摘要统计数据的显著性水平。默认为 5e-08。

例子

```
my_file <- "my_file.csv"
read_easy(my_file, pval = 1e-06)
```

返回值

该函数返回摘要统计数据的数据框。

get_eaf_from_1000G

介绍

从1000G的MAF文件中提取EAF并将其与输入数据匹配。

用法

```
get_eaf_from_1000G(dat, path, type = "exposure")
```

参数

- `dat`：一个数据框，为TwoSampleMR包格式的数据
- `path`：一个字符串，表示包含1000G MAF文件的目录路径。
- `type`：一个字符串，表示数据是“exposure”（暴露因素）还是“outcome”（结果），默认为“exposure”。

值

一个数据框，其中包含输入数据的EAF和类型信息（原始、修正或错误）。

详细说明

该函数将读取1000G MAF文件，然后将其与输入数据进行匹配。对于每个SNP，该函数将检查输入数据中的效应等位基因与1000G中的效应等位基因是否匹配。

如果不匹配，则将EAF设置为NA并将其类型设置为“error”。对于匹配的SNP，EAF将设置为1000G MAF中的值（如果输入数据的效应等位基因是minor allele），或1-MAF（如果输入数据的效应等位基因是major allele），并将其类型设置为“raw”或“corrected”。

如果`type`参数设置为“outcome”，则函数将使用输入数据中的结果等位基因来查找EAF。

在处理完所有SNP后，该函数将输出一些有关匹配成功、失败以及NA的信息，以及类型信息的说明。

示例

以下是使用该函数的示例：

```
dat <- get_eaf_from_1000G(dat, "S:/GWAS数据/本地LD依赖文件/fileFrequency.frq", type = "exposure")

# 检查输出
head(dat)
```

`fileFrequency.frq`为PLINK1.9输出的，根据1000G数据提取的MAF数据

1000G参考文件下载地址：http://fileserve.mrcieu.ac.uk/ld/1kg.v3.tgz

可自行提取MAF数据。或从`GetScience`公众号中获取已经提取好的`fileFrequency.frq`文件

get_chr_pos

该函数利用MungeSumstats包匹配rsid的染色体位置。

用法

```
get_chr_pos(dat, type = "exposure")
```

参数

- `dat`：一个数据框，TwoSampleMR格式
- `type`：一个字符串，表示要获取SNP染色体位置和参考序列的SNP类型。可选值为"exposure"或"outcome"。

返回值

一个数据框，其中包含输入数据框的信息，以及新列`chr.exposure`或`chr.outcome`，表示每个SNP的染色体编号。新列`pos.exposure`或`pos.outcome`表示每个SNP在染色体上的位置。

函数说明

该函数使用`format_sumstats`和`format_data`函数从1000G项目中获取SNP的染色体位置和参考序列信息。

示例

以下示例演示如何使用`get_chr_pos`函数：

```
# 获取曝露变量SNP的染色体位置和参考序列
exposure_chr_pos <- get_chr_pos(dat, type = "exposure")

# 获取结果变量SNP的染色体位置和参考序列
outcome_chr_pos <- get_chr_pos(dat, type = "outcome")
```

get_f函数

描述

`get_f`函数计算样本的F统计量并返回F值大于指定阈值的样本数据。返回的数据包括每个SNP的R2和F值

用法

```
get_f(dat, F_value = 10)
```

参数

- `dat`: TwoSampleMR格式，一定要包含`eaf.exposure`，`beta.exposure`，`se.exposure`，和`samplesize.exposure`的数据框。
- `F_value`: 指定的F统计量的阈值，F值大于该阈值的样本将被返回。默认值为10。

注，本计算公式为

F值：$\frac{r^2(N-2)}{(1-r^2)}$ R2值：

$$\frac{2\hat{\beta}^2 MAF(1-MAF)}{2\hat{\beta}^2 MAF(1-MAF) + (\mathrm{se}(\hat{\beta}))^2 2NMAF(1-MAF)}.$$

公式参考文献：

A Multivariate Genome-Wide Association Analysis of 10 LDL Subfractions, and Their Response to Statin Treatment, in 1868 Caucasians - PMC (nih.gov)

Large-scale association analyses identify host factors influencing human gut microbiome composition - PMC (nih.gov)

mr_dircreate_base

描述

`mr_dircreate_base`函数创建基本的目录结构以保存MR分析结果和图形。

用法

```
mr_dircreate_base(root_dir, project_name, date = NULL)
```

参数

- `root_dir`: 保存结果文件夹的根目录。
- `project_name`: 项目名称，用于在根目录下创建一个以此命名的子目录。
- `date`: 日期（可选），用于在子目录名称中添加日期以区分不同日期的结果文件夹。默认为`NULL`，表示不添加日期。

值

`mr_dircreate_base`函数返回一个包含结果文件夹路径的列表。

示例

```
# 创建结果文件夹
res_dir <- mr_dircreate_base("path/to/root/dir", "project_name", date =
"20220327")

# 打印结果文件夹路径
print(res_dir)
```

注意事项

- `root_dir`参数指定保存结果文件夹的根目录。
- `project_name`参数指定项目名称，用于在根目录下创建一个以此命名的子目录。
- `date`参数（可选）用于在子目录名称中添加日期以区分不同日期的结果文件夹。默认为`NULL`，表示不添加日期。
- `mr_dircreate_base`函数将在根目录下创建一个名为`project_name`的子目录，并在该子目录下创建4个子目录，分别命名为`1.figure`、`2.table`、`3.figure of sig res`和`4.snp with Fval`。函数返回一个包含结果文件夹路径的列表。

clean_expo

描述

用于快捷筛选工具变量的函数，可执行P值筛选，EAF值筛选，clump。

用法

```
## 完整可选参数
clean_expo(expo, pval, low_af = 0.5, high_af = 0.5, clump = TRUE, kb = 10000, r2
= 0.001, LD_file = NULL, af_filter = FALSE)

##不提供LD_file则自动在线clump
clean_expo(expo, pval, clump = TRUE, kb = 10000, r2 = 0.001)
##提供则本地clump
clean_expo(expo, pval, clump = TRUE, kb = 10000, r2 = 0.001, LD_file=LD_file)
```

参数

- `expo`: 一个数据框，其中包含遗传暴露指标的SNP名称、beta值、标准误、p值和频率。
- `pval`: 用于筛选遗传暴露指标的p值阈值。p值小于此阈值的SNP将被保留。
- `low_af`: 频率过滤的下限值。默认为0.5。如果`af_filter`为TRUE，则只有遗传暴露指标的频率低于此值或高于`high_af`时，才会被保留。
- `high_af`: 频率过滤的上限值。默认为0.5。
- `clump`: 一个逻辑值，指示是否使用PLINK进行SNP聚类。默认为TRUE。
- `kb`: 聚类的窗口大小（以kb为单位）。默认为10000。
- `r2`: LD阈值。默认为0.001。
- `LD_file`: PLINK二进制文件的路径。如果未提供，则默认在线clump。
- `af_filter`: 一个逻辑值，指示是否启用频率过滤。默认为FALSE。

clean_list

描述

用于清理列表中元素行数的R包。常用于批量化质量控制。

用法

```
clean_list(list, nrow = 10)
```

参数

- `list`: 一个列表，其中包含多个元素。
- `nrow`: 用于筛选元素的行数阈值。如果元素的行数小于此阈值，则该元素将被删除。默认为10。

返回值

一个列表，其中仅包含行数大于 `nrow` 的元素。

例子

```r
# 创建一个包含5个数据框的列表，每个数据框包含1-5行
set.seed(123)
lst <- list(data.frame(a = rnorm(1), b = rnorm(1)),
            data.frame(a = rnorm(2), b = rnorm(2)),
            data.frame(a = rnorm(3), b = rnorm(3)),
            data.frame(a = rnorm(4), b = rnorm(4)),
            data.frame(a = rnorm(5), b = rnorm(5)))

# 运行 clean_list 函数，将阈值设置为2
cleaned_lst <- clean_list(lst, nrow = 3)

# 查看清理后的列表
cleaned_lst
```

clean_IV_from_outsig

用于从一个数据框中清理具有显著的MR反向因果效应P值的IV。

用法

```r
clean_IV_from_outsig(dat, MR_reverse = 1e-03)
```

参数

- `dat`: 一个数据框，其中包含每个IV和其与结果变量之间的MR反向因果效应的P值。
- `MR_reverse`: 用于筛选IV的MR反向因果效应P值阈值。具有P值小于此阈值的IV将被保留。默认为1e-03。

返回值

一个数据框，其中包含P值大于 `MR_reverse` 值的IV（也就是反向不显著的IVs）。

作者信息

- 代码作者：广州医科大学 第一临床学院 周浩彬　第二临床学院 谢治鑫
- 帮助文档作者： 周浩彬

- 时间：2023/3/27
- 适配版本: Get_MR1.0

- 公众号： GetScience

- 致谢：感谢广州医科大学 第六临床学院 黄覃耀和 南山学院 林子凯在孟德尔随机化概念，代码思路等提供的重要的建设性建议。