(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 112359124 A
(43)申请公布日 2021.02.12

(21)申请号 202011244166.X

(22)申请日 2020.11.10

(71)申请人 沈阳农业大学
地址 110866 辽宁省沈阳市沈河区东陵路120号

(72)发明人 张敖 张学才 崔震海 阮燕晔 樊金娟 董小妹 陈珊 史忠贤 孙凯悦 张立军

(74)专利代理机构 大连智高专利事务所(特殊普通合伙) 21235
代理人 张钦

(51)Int.Cl.
C12Q 1/6895(2018.01)
G16B 20/20(2019.01)

权利要求书1页 说明书20页 附图1页

(54)发明名称

一种依据植物双亲基因型信息虚拟合成杂种基因型的方法及应用

(57)摘要

本发明属于植物分子辅助育种领域，公开了一种依据植物双亲基因型信息虚拟合成杂种基因型的方法及应用。包括各类作物和野生植物的父、母亲本(包括自交系、无性系、杂交种、农家品种和天然混交群体)基因组信息(包括DNA全序列数据和基因分型数据，包括来自芯片测序平台数据和简易基因组测序平台GBS的数据)，利用生物信息学技术推测或虚拟合成杂交种F1代基因型的方法。解决了在利用全基因组预测技术预测自交系一般配合力、特殊配合力和杂交组合表现型时没有基因型信息可用的问题。

CN 112359124 A

1.一种依据植物双亲基因型信息虚拟合成杂种基因型的方法，其特征是，对原始基因组数据进行质量控制、去劣补遗、然后将基因组数据中的碱基符号转变为代表显性的A和隐性的a，再将原始的二倍体形式基因型转变为四倍体形式的基因型，然后将其数值化处理，最后用父、母亲本数值化的四倍体形式的基因型虚拟合成杂交种的二倍体基因型；所获得的数值化的杂交种二倍体基因型数据可直接代入全基因组预测模型，用来预测自交系的特殊配合力和杂交组合的性状表型。

2.如权利要求1所述的一种依据植物双亲基因型信息虚拟合成杂种基因型的方法，其特征是，包括以下步骤：

(1)将来自父、母亲本的基因型数据，进行质量控制，去除杂合率、最小等位基因频率不符合实际情况的材料和SNP标记，去掉缺失率高的分子标记；

(2)补充遗缺数据；

(3)计算每个碱基座的等位碱基频率；

(4)根据等位碱基频率将碱基替换为A/a形式；

(5)将步骤4得到的A/a形式的基因型分别归入父、母本材料列表；

(6)从父、母本材料基因型列表中抽取基因型，合成四碱基A/a形式的杂交种基因型；

(7)根据A和a的频率将A/a转换成数值；

(8)虚拟合成数值化的二倍体杂交种基因型；

(9)将虚拟合成数值化的二倍体杂交种基因型数据直接代入全基因组预测模型预测自交系特殊配合力或杂交组合的表现型。

一种依据植物双亲基因型信息虚拟合成杂种基因型的方法及应用

技术领域

[0001] 本发明涉及植物分子辅助育种领域，涉及一种依据植物双亲基因型信息虚拟合成杂种基因型的方法及应用；具体涉及利用生物信息方法根据植物父、母本双亲的基因型信息预测杂种一代(F1)基因型的方法。

背景技术

[0002] 植物杂交种(hybrid)是由基因型不同父、母亲本自交系之间进行杂交产生的第一代种子(通常称为杂种F1代)。杂交种在生长势、抗逆性和产量方面优于父、母亲本，即，产生杂种优势。然而，不是任何具有遗传差异(基因型不同)的自交系之间杂交都能产生适合农业产生需要的优良杂交种。产生优良杂交种的两个自交系之间不仅要有遗传差异，而且还要优良性状互补。所以，杂交育种者不仅要培育能够产生杂种优势的优良自交系，还要进行自交系之间的杂交组合筛选，从中获得优良杂交种。

[0003] 传统的优良自交系选择和杂交组合筛选过程除需要大量土地外，还需要大量的田间管理和性状测定；随着育种进程，育种单位的自交系和杂交组合会越来越多，需要的投入也不断增加，成本增大。此外，目前的植物杂交种在产量、抗性和品质方面已经达到较高水平，单纯依靠传统杂交育种方法将难有大的突破性提高。

[0004] 随着测序技术的发展，基因测序成本大幅度降低，分子辅助育种技术已经发展成为可以实用的技术。将分子辅助育种技术与传统杂交技术结合，将是未来降低杂交育种成本，提高育种效率，提高育种水平的必然途径。分子辅助育种技术包括分子标记辅助育种技术(molecular marker assistance)和全基因组预测技术(genomic prediction)。

[0005] 分子标记辅助育种技术(molecular marker assistance)利用已知的与特定表型性状相关联的DNA分子标记，不需要经过田间种植和性状测定，直接对欲选择的自交系或杂交种进行分子标记筛选，将不含有特定分子标记的材料淘汰，可以节省大量的人力、物力和时间，提高育种效率。但是，与特定表型性状关联的DNA分子标记会因植物群体的遗传背景变化而变化，没有广泛的通用性。因此，对于没有进行过分子标记研究的植物群体，将无法应用分子标记辅助技术。

[0006] 全基因组预测(genomic prediction)技术，与分子标记辅助育种技术不同，不依赖于已知的分子标记，而是根据待预测材料的基因型，利用已经建立好的基因型与实测表现型关系的数学模型，来预测表现型，将不符合表现型要求的材料淘汰。这种方法将减少大量的田间种植和性状测定工作，可以节省大量的人力、物力和时间，提高育种效率。与分子标记辅助育种相比，全基因组预测的优势是不需要事先知道表型性状的分子标记，只需知道被预测材料的基因型即可，尤其适合由多基因控制的数量性状。因此，与分子标记辅助技术相比，全基因组预测技术适用性更加广泛。

[0007] 全基因组预测技术即可用来预测自交系的一般配合力(general combining ability,GCA)特殊配合力(specific combining ability,SCA)，也可用来预测杂交组合的

性状表现，作为筛选优良杂交组合的依据。某个自交系的GCA是自交系与许多其他自交系(测验种)杂交都产生具有强杂种优势F1的能力；特殊配合力是指自交系与特定遗传背景自交系进行杂交产生强杂种优势的能力。一般配合力和特殊配合力是衡量优良自交系的重要指标。自交系的一般配合力、特殊配合力和杂交组合的表型并不决定于单一亲本自交系的基因型，而是决定于两个亲本自交系基因型的组合。因此，利用全基因组预测技术预测自交系的特殊配合力或杂交组合的性状表现，都需要了解杂交组合的基因型。

[0008] 然而，全基因组预测技术不管是用来预测自交系的一般配合力和特殊配合力，还是用来预测杂交组合的表现型时，都面临一个没有被预测杂交组合基因型信息的问题。由于在进行全基因组预测时，研究者手里掌握的信息是：全部自交系的基因型，部分杂交组合的表现型，如果不计成本的话，还可以实测这部分杂交组合的基因型。但是，没有待预测杂交组合的基因型，即使不计成本肯花钱也得不到。因为，对于待预测杂交组合，由于没有做真实的杂交，也就没有真实的杂种一代种子用来基因测序。没有待预测杂交组合的基因型，也就不能利用全基因组预测技术高精度的预测自交系的一般配合力、特殊配合力和杂交组合的表现，从而影响了全基因组预测技术在实际育种中的应用。

[0009] 综上，没有待预测杂交组合的基因型信息，是限制全基因组预测技术在杂交育种中应用的关键障碍。因此，解决问题的关键是必须设法获得待预测杂交组合的基因型信息。单个杂交种的测序费用高于自交系的费用，且杂交组合数量远大于其亲本的数量。解决问题的有效方法是利用父、母亲本自交系基因型信息推测或虚拟合成杂交组合的基因型，目前尚未见到这方面的技术。

[0010] 需要指出的是，利用父、母亲本自交系基因型信息虚拟合成杂交组合基因型，并不是父、母本自交系基因型数据的简单相加，因为两个自交系基因组重新组合后，某些等位基因位点将会出现杂合性，更为复杂的是为加快自交系选育进度，实践上都是在高代进行测配选择，自交系的很多等位基因还未纯合，这些情况在基因型虚拟合成时必须加以考虑。另外，在一些特殊情况下，育种者会用自交系与杂交种杂交，所产生后代基因型呈现多样化，虚拟合成将会更加复杂。

[0011] 此外，植物的基因型信息分为DNA全序列信息和基因分型信息：DNA全序列信息反映的是核苷酸在DNA分子上的精确排列；基因分型信息反映的是DNA分子的单核苷酸多态性(SNP，single nucleotide polymorphism)，并不是完整的、精确的核苷酸在DNA上的排列信息。因此，在进行杂交组合基因型虚拟合成时，需要对不同形式的基因组信息进行预处理。

发明内容

[0012] 为解决利用全基因组预测技术预测自交系特殊配合力和杂交组合表现型时缺乏杂交组合基因型的问题，本发明提供一种依据植物，包括各类作物和野生植物的父、母亲本(包括自交系、无性系、杂交种、农家品种和天然混交群体)基因组信息(包括DNA全序列数据和基因分型数据，包括来自芯片测序平台数据和简易基因组测序平台GBS的数据)，利用生物信息学技术推测或虚拟合成杂交种F1代基因型的方法。首先对原始基因组数据进行质量控制、去劣补遗、然后将基因组数据中的碱基符号转变为代表显性的A和隐性的a，再将原始的二倍体形式基因型转变为四倍体形式的基因型，然后将其数值化处理，最后用父、母亲本数值化的四倍体形式的基因型虚拟合成杂交种的二倍体基因型。所获得的数值化的杂交种

二倍体基因型数据可直接代入全基因组预测模型，用来预测自交系的特殊配合力和杂交组合的性状表型。

[0013] 本发明的技术方案如下：

[0014] 依据父、母亲本基因组信息利用生物信息学技术虚拟合成杂交种F1代基因型的方法，包括以下步骤(图1)：

[0015] (1)将来自父、母亲本的基因型数据，进行质量控制，去除杂合率、最小等位基因频率不符合实际情况的材料和SNP标记，去掉缺失率高的分子标记。

[0016] (2)补充遗缺数据。

[0017] (3)计算每个碱基座的等位碱基频率。

[0018] (4)根据等位碱基频率将碱基替换为A/a形式，等位基因频率高的为A，小的为a。如：在某个碱基座上，有A和T，A的等位基因频率>0.5，则A为A，T为a，反之T为A，A为a。

[0019] (5)将步骤4得到的A/a形式的基因型分别归入父、母本材料列表。

[0020] (6)从父、母本材料基因型列表中抽取基因型，合成四碱基A/a形式的杂交种基因型。

[0021] (7)根据某个材料在特定基因座上A和a的频率，将所有A/a形式的基因型信息转换成数值，例AAAA为1，AAAa为0.75，AAaa为0.5，Aaaa为0.25，aaaa为0。这些值恰好等于二倍体杂交种在特定碱基座上的频率值，具有很好的代表性。

[0022] (8)将虚拟合成数值化的二倍体杂交种基因型数据直接代入全基因组预测模型预测自交系特殊配合力或杂交组合的表现型。

[0023] 上述一种依据植物双亲基因型信息虚拟合成杂种基因型在作物育种上的应用。

[0024] 作物育种的最终目标是组配高产、稳产、高质的杂交种。一般情况下，育种单位会将所有的自交系按照系谱或血缘分成两个或多个群体，再将不同群体之间进行杂交。由于群体数量过于庞大，例如按照比较小的规模计算，A群100个材料，B群100个材料，则有10000个杂交组合，把10000个杂交组合全部种在试验田，重复2年，每年3个地点，每个地点2个重复，则需要种植120000行，占用大量土地资源，同时人工表型鉴定的费用也非常巨大。利用虚拟杂交种基因型拟合技术可以选取有代表性的两个群体各20个材料建模，结合全基因组预测技术预测其他所有杂交组合的表现(育种值)，则只有400个组合，总体田间工作量减少100%-(400×2×3×2)÷120000×100%＝96%。

[0025] 本发明的有益效果如下：首次依据植物(包括作物和野生植物)父、母亲本(包括自交系和杂交种)的基因型数据，包括DNA全序列数据和基因分型数据，包括来自芯片测序数据和简易基因组(GBS)的测序数据，利用生物信息学技术推测或虚拟合成杂交种F1代的基因型，解决了在利用全基因组预测技术预测自交系一般配合力、特殊配合力和杂交组合表现型时没有基因型信息可用的问题，破除了限制全基因组预测技术在杂交育种中应用的关键障碍；虚拟合成的基因型还可用来分析杂交组合的基因组结构，指导制定杂交组合的选配方案。总之，该方法的应用将有助于提高杂交育种的效率和水平。

附图说明

[0026] 图1依据父、母双亲基因型信息虚拟合成杂交种基因型的技术路线图。

具体实施方式

[0027] 下面结合具体实施例对本发明做进一步的说明，若无特殊说明，本发明所用技术或软件均为本领域常规应用技术或软件。

[0028] 实施例1

[0029] 依据父、母亲本基因组全基因组精确序列信息利用生物信息学技术虚拟合成杂交种F1代基因型的方法，包括以下步骤：

[0030] (1)将来自父、母亲本的测序原始数据利用SNP Calling方法获取单核苷酸多态性(SNP)信息并转换成hapmap格式(*.hmp.txt)。

[0031] (2)用TASSEL软件进行质量控制。

[0032] (3)杂合基因型的合成。

[0033] a)下载并安装R语言和RStudio(略)。本示例在Windows10下进行。

[0034] b)利用下面的命令依次选择Hapmap格式的文件和表型文件，表型文件的目的是提供父本和母本信息，表型文件是3列的文本文件，第一列的列名为LINE，第二列为TESTER，第三列为表型性状的名称。本程序除了读取文件外，也对文件进行检测，看是否符合规范，除此之外，根据表型文件所在目录，建立生成目录。

[0035]

```
if (!exists("myGenoName")){
   cat("请选择基因型文件...\n Please choose a genotipic file in HMP format...\n")
   myGenoName <- choose.files()
}else{
   if (length(myGenoName)==0){
      cat("请选择基因型文件...\n Please choose a genotipic file in HMP format...\n")
      myGenoName <- choose.files()
   }
}
if (!exists("myPhenoName")){
      cat("请选择表型文件...\n Please choose a phenotipic file in TXT format...\n")
      myPhenoName <- choose.files()
}else{
   if (length(myPhenoName)==0){
      cat("请选择表型文件...\n Please choose a phenotipic file in TXT format...\n")
      myPhenoName <- choose.files()
   }
}
myGeno <- readFiles(header=FALSE,fname=myGenoName)      # Import the Genotipic file.


## 检查基因型数据是否正确
```

[0036]

```
bit=nchar(as.character(myGeno[2,12]))
if(!bit==1) {stop("Data format is hapmap with haploid allele! Please Check Genotype data!")}

myPheno <- readFiles(header=TRUE,fname=myPhenoName)  # Import the Phenotipic file.

if (length(which(is.na(myPheno$TESTER)==TRUE))==nrow(myPheno)) {
  noTester <- TRUE
}else{
  noTester <- FALSE
  # 判断 TESTER 是否为杂交种,并分开杂交种
  if (length(grep("@",myPheno$TESTER))>0){
    Tester0 <- strsplit(myPheno$TESTER[grep("@",myPheno$TESTER)],split="@")
    Tester1 <- NULL
    Tester2 <- NULL
    for (li in 1:length(Tester0)){
      Tester1 <- c(Tester1,Tester0[[li]][1])
      Tester2 <- c(Tester2,Tester0[[li]][2])
    }
    Tester1 <- unique(Tester1)
    Tester2 <- unique(Tester2)
    Testerx <- myPheno$TESTER[-c(grep("@",myPheno$TESTER))]
    TESTERS <- unique(c(Tester1,Tester2,Testerx))
  }else{
    TESTERS <- unique(c(myPheno$TESTER))
  }
}
```

[0037]

```
# 判断 Line 是否为杂交种,并分开杂交种
if (length(grep("@",myPheno$LINE))>0){
   Line0 <- strsplit(myPheno$LINE[grep("@",myPheno$LINE)],split="@")
   Line1 <- NULL
   Line2 <- NULL
   for (li in 1:length(Line0)){
      Line1 <- c(Line1,Line0[[li]][1])
      Line2 <- c(Line2,Line0[[li]][2])
   }
   Line1 <- unique(Line1)
   Line2 <- unique(Line2)
   Linex <- myPheno$LINE[-c(grep("@",myPheno$LINE))]
   LINES <- unique(c(Line1,Line2,Linex))
}else{
   LINES <- unique(c(myPheno$LINE))
}

## 获取性状名称
index <- names(myPheno) %in%
c("LOC","ENV","LINE","TESTER","YEAR","P1P2")
fstName <- names(myPheno)[!index]     # The trait name.

## 自动设置输出目录
newDirectory <- paste0(dirname(myPhenoName),"/","GenoInfo")
if(!dir.exists(newDirectory)){
   dir.create(newDirectory)
}
setwd(newDirectory)
cat("Output Directory:",getwd(),"\n")
```

[0038] c)表型和基因型数据经常会由于漏测或结果太差等原因引起数量不等,因此需要先对表型和基因型进行对应,用如下命令。

```
myGenoInformation <- myGeno[,1:11]    # 基因型数据基本信息
myGenoMarkers <- myGeno[,12:ncol(myGeno)]    # 基因型数据标记信息
myGenoIndividuals <- as.character(myGenoMarkers[1,])    # 提取所有的材料名
myGenoMarkers <- myGenoMarkers[,order(myGenoIndividuals)]    # 按照材料名排序标记信息部分

## 基因型与表型材料名称对应
if (noTester==T){
  myPhenoN <- LINES
}else{
  myPhenoN <- unique(c(LINES,TESTERS))
}
myGenoN <- unique(myGenoIndividuals)
bothGP <- intersect(myPhenoN,myGenoN)
### 有表型的基因型数据
index <- myGenoIndividuals %in% bothGP
cat("Unmatched genotype: \n")
print(unique(myGenoIndividuals[!index]))
myGenoMarkers <- as.matrix(myGenoMarkers[,index])
myGenoIndividuals <- as.character(myGenoMarkers[1,])    # 提取所有的材料名
myGenoMarkers <- myGenoMarkers[,order(myGenoIndividuals)]    # 按照材料名排序标记信息部分
myGeno <- cbind(myGenoInformation,myGenoMarkers)    # 重新合成基因型数据（排序后）
#myGenoIndividualsT <- t(myGeno[1,-(1:11)])    # 获得排序后的材料名（包括Lines 和 Testers（亲本 1 或亲本 2））
### 有基因型的表型数据
#### LINE
if (length(grep("@",myPheno$LINE))){
```

[0039]

```
  Line00 <- strsplit(myPheno$LINE,split="@")
  index <- myPheno$LINE %in% bothGP    # 查看 Line 中基因型和表型的对
应索引(非杂交种的匹配情况)
  for (li in grep("@",myPheno$LINE)){
    index[li] <- all((Line00[[li]] %in% bothGP)==TRUE)    # 修正杂交种的匹
配情况
  }
  cat("Unmatched Line: \n")
  print(unique(myPheno$LINE[!index]))    # 查看是否有未匹配的项并列出
}else{
  index <- myPheno$LINE %in% bothGP    # 查看 Line 中基因型和表型的对
应索引
  cat("Unmatched Line: \n")
  print(unique(myPheno$LINE[!index]))    # 查看是否有未匹配的项并列出
}
```

[0040]

```
myPheno <- myPheno[index,]    # 去掉未对应表型数据
#### TESTER
if (noTester==F){
  if (length(grep("@",myPheno$TESTER))){
    Tester00 <- strsplit(myPheno$TESTER,split="@")
    index <- myPheno$TESTER %in% bothGP    # 表型 Tester 中基因型和表
型的对应索引
    for (li in grep("@",myPheno$TESTER)){
      index[li] <- all((Tester00[[li]] %in% bothGP)==TRUE)    # 修正杂交种
的匹配情况
    }
    cat("Unmatched Line: \n")
    print(unique(myPheno$LINE[!index]))    # 查看是否有未匹配的项并列出
  }else{
    index <- myPheno$TESTER %in% bothGP    # 表型 Tester 中基因型和表
```

[0041]

```
型的对应索引
    cat("Unmatched Tester: \n")
    print(unique(myPheno$TESTER[!index]))    # 查看是否有未匹配的项并
列出
  }
  myPheno <- myPheno[index,]    # 去掉未对应的表型数据
}

if (myGeno[1,1]=="rs#"){
  names(myGeno) <- myGeno[1,]    # 为数据框添加标题
  myGeno <- myGeno[-1,]    # 去掉第一行
}

GD <- as.matrix(myGeno[,-(1:11)])    # get genotype matrix    获得基因型矩阵
```

[0042] d)将Hapmap格式中的“N”替换成缺失值“NA”,用于后续分析,使用如下命令。

[0043]

```
## "N"换成 NA
if (any(GD=="N")){
  GD <- sub("N",NA,GD)
}
myGeno[,12:ncol(myGeno)] <- GD    # 将 myGeno 替换成有 NA 形式
Marker <- myGeno[,c(1,3,4)]    # 获得标记信息数据框
colnames(Marker) <- c("SNP","Chromosome","Position")    # 修改标记数据框
的列名，对应相应数据
rownames(GD) <- Marker$SNP    # 增加标记名称
No.markers <- character()
No.markers["original"] <- nrow(GD)
```

[0044] e)没有多态性的亲本合成没有意义,因此利用下面命令去除没有多态性的亲本。

[0045]

```
polymorphism <- TRUE

#! 删除无多态性的标记
```

[0046]

```
delet_monomorphic <- function(myVector){
  monomorphic <- numeric()
  if(length(as.numeric(table(myVector)))<=1)
  {
    monomorphic <- 1     # 1 表示无多态性
  }else{
    monomorphic <- 0    # 0 表示有多态性
  }
  return(monomorphic)
}

## 多态性检测
if (polymorphism == TRUE){
  cat("Status: Deleting no polymorphism loci!\n")
  system.time(index_monomorphic <- apply(GD,1,delet_monomorphic))    # 获得无多态性标记索引
  myGeno <- myGeno[index_monomorphic==0,]
  GD <- GD[index_monomorphic==0,]
  Marker <- Marker[index_monomorphic==0,]     # 获得标记信息数据框
  rm(index_monomorphic)
}
```

[0047] f)Hapmap格式中,用IUPAC(International Union of Pure and Applied Chemistry,国际纯粹与应用化学联合会)命名法来表示二倍体碱基,需要还原成二倍体形式。同时,利用table()函数获得每个碱基的频率,将频率高的碱基转换为“AA”,频率低的碱基转换为“aa”,杂合转换为“Aa”,用如下命令:

[0048]

```
#! 功能函数：变为双碱基形式
douobleBase <- function(x){
  tema <- x
  if (!length(which(is.na(tema)==T))==length(tema)){
    tema <- sub("A","AA",tema)
```

[0049]

```
tema <- sub("T","TT",tema)
tema <- sub("C","CC",tema)
tema <- sub("G","GG",tema)
tema <- sub("R","AG",tema)
tema <- sub("Y","CT",tema)
tema <- sub("S","CG",tema)
tema <- sub("W","AT",tema)
tema <- sub("K","GT",tema)
tema <- sub("M","AC",tema)
b <- tema
b[which(is.na(tema))] <- ""     # NA 替换成""
b <- paste(b,collapse = "")     # 合并字母
b <- unlist(strsplit(b,split=""))     # 拆分字母
cb <- as.data.frame(table(b))     # 获取字母个数
cb <- cb[order(cb[,2],decreasing = TRUE),]     # 倒序排列
A <- as.character(cb[1,1])
a <- as.character(cb[2,1])
tema <- gsub(A,1,tema)
tema <- gsub(a,0,tema)
if (nrow(cb)>2){
   tema <- gsub("[^1|^0]",NA,tema)
}
tema <- gsub(1,"A",tema)
tema <- gsub(0,"a",tema)
for(xv in 1:length(tema)){
   if (!is.na(tema[xv])){
      temb <- unlist(strsplit(tema[xv],split=""))
      temb <- sort(temb,decreasing=T)
      tema[xv] <- paste(temb,collapse = "")
   }else{
```

[0050]

```
            tema[xv] <- NA
          }
        }
      }
    return(tema)
}
## 单碱基变双碱基
No.markers["polymorphism"] <- nrow(GD)    # 获得筛选后的标记数量
cat("Getting genotype of doube-base type: \n")
system.time(GD_D <- apply(GD,1,douobleBase))    # 单碱基变双碱基
GD_D <- t(as.matrix(GD_D))    # 转置，所有分析基于该矩阵
```

[0051] g)缺失值可能会对结果造成影响,使用概率补缺失方案,随机生成数字,该数字落在哪个概率分布上,则将缺失值补充为该概率对应的碱基(此步骤非必须),命令如下。

[0052]

```
### 概率方案补缺失
if (Geno_Probability_imputation==TRUE){
  cat("Status: Imputating!\n")
  system.time(impu_prob_value <- apply(GD_D,1,probability_impute))
  GD_D <- t(impu_prob_value)
}
```

[0053] h)杂交种四倍体基因型虚拟合成,如,AA和aa合成为AAaa,而Aa和aa则合成为Aaaa,以此类推。命令如下:

[0054]

```
#! 功能函数：杂交种基因型合成
Synthetic_hybrids <- function(x){
  if (length(grep("@@",x))>0){
    xx <- strsplit(x,split="@@")
    if (Synthetic_ignore_gene==TRUE){
      x1 <- GD_D[,xx[[1]][1]]
      x2 <- GD_D[,xx[[1]][2]]
      if (any(is.na(x1)==TRUE)){
```

[0055]

```
            x1[which(is.na(x1)==TRUE)] <- ""
          }
          if (any(is.na(x2)==TRUE)){
            x2[which(is.na(x2)==TRUE)] <- ""
          }
          xresu <- paste0(x1,x2)
        }else{
          xresu <- paste0(GD_D[,xx[[1]][1]],GD_D[,xx[[1]][2]])
          xresu <- gsub("NA",NA,xresu)
        }
      }else{
        xx <- strsplit(x,split="@")
        if (Synthetic_ignore_gene==TRUE){
          x1 <- GD_D[,xx[[1]][1]]
          x2 <- GD_D[,xx[[1]][2]]
          if (any(is.na(x1)==TRUE)){
            x1[which(is.na(x1)==TRUE)] <- ""
          }
          if (any(is.na(x2)==TRUE)){
            x2[which(is.na(x2)==TRUE)] <- ""
          }
          xresu <- paste0(x1,x2)
        }else{
          xresu <- paste0(GD_D[,xx[[1]][1]],GD_D[,xx[[1]][2]])
          xresu <- gsub("NA",NA,xresu)
        }
      }
      return(xresu)
}
#! 功能函数，字母排序联用函数 1
```

[0056]

```
Aasort1 <- function(Aav){
  Aav <- as.matrix(Aav)
  sAa <- apply(Aav,1,Aasort2)
  return(sAa)
}
#! 功能函数，字母排序联用函数 2
Aasort2 <- function (Aa) {
  if (!is.na(Aa)){
    Aax <- unlist(strsplit(Aa,split=""))
    Aax <- sort(Aax,decreasing=T)
    Aax <- paste(Aax,collapse = "")
  }else{
    Aax <- NA
  }
  return(Aax)
}
### 杂交种合成
Synthetic_ignore_gene <- TRUE
## Line 的杂交种合成
if (length(grep("@",myPheno$LINE))>0){
  HLine <- as.matrix(myPheno$LINE[grep("@",myPheno$LINE)])
  system.time(HLGeno <- apply(HLine,1,Synthetic_hybrids))  # 获得无多态性
标记索引
  rownames(HLGeno) <- rownames(GD_D)
  colnames(HLGeno) <- HLine[,1]
  system.time(test <- apply(HLGeno,1,Aasort1))
  HLGeno <- t(test)
  if (nrow(HLGeno)==1){
    HLGeno <- t(HLGeno)
    colnames(HLGeno) <- HLine[,1]
```

[0057]

```
    }
    if (!HLine[1,1] %in% colnames(GD_D)){
        GD_D <- cbind(GD_D,HLGeno)
    }
}
## Tester 的杂交种合成
if (length(grep("@",myPheno$TESTER))>0){
    HTester <- as.matrix(myPheno$TESTER[grep("@",myPheno$TESTER)])
    system.time(HTGeno <- apply(HTester,1,Synthetic_hybrids))   # 获得无多态
性标记索引
    rownames(HTGeno) <- rownames(GD_D)
    colnames(HTGeno) <- HTester[,1]
    system.time(test <- apply(HTGeno,1,Aasort1))
    HTGeno <- t(test)
    if (nrow(HTGeno)==1){
        HTGeno <- t(HTGeno)     # 只有一个时转换
        colnames(HTGeno) <- HTester[,1]
    }
    if (!HTester[1,1] %in% colnames(GD_D)){
        GD_D <- cbind(GD_D,HTGeno)
    }
}
### Line×Tester 合成
if (noTester==F){
    HH <- as.matrix(paste0(myPheno$LINE,"@@",myPheno$TESTER))
    myPheno$LINExTESTER <- HH[,1]
    system.time(HHGeno <- apply(HH,1,Synthetic_hybrids))
    rownames(HHGeno) <- rownames(GD_D)
    colnames(HHGeno) <- HH[,1]
    system.time(test <- apply(HHGeno,1,Aasort1))
```

[0058]

```
    HHGeno <- t(test)
    if (nrow(HHGeno)==1){
      HHGeno <- t(HHGeno)
      colnames(HHGeno) <- HLine[,1]
    }
    if (!HHGeno[1,1] %in% colnames(GD_D)){
      GD_D <- cbind(GD_D,HHGeno)
    }
}
```

[0059] i)将四倍体基因型数据转换成数值型,以便用于进一步的全基因组预测分析,命令如下。

[0060]

```
#! 功能函数：杂交种基因型数值化联用函数 1
numericH <- function(x){
  x <- as.matrix(x)
  nHv <- apply(x,1,numericH2)
  return(nHv)
}

#! 功能函数：杂交种基因型数值化联用函数 1(单个值，每个 A 代表的值)
numericH2 <- function(Hx){
  if (!is.na(Hx)){
    xmax <- nchar(max(Hx,na.rm=T))
    eachA <- 1/xmax
    eachCell <- unlist(strsplit(Hx,split=""))
    countA <- length(which(eachCell=="A"))
    Hvalue <- countA*eachA
  }else{
    Hvalue <- Hx
  }
  Hvalue <- as.numeric(Hvalue)
```

[0061]

```
  return(Hvalue)
}
### 杂交种基因型数值化
GD_DN <- apply(GD_D,2,numericH)
```

[0062] j)生成数值型基因型,用如下代码。程序会在表型数据目录下建立的GenoInfo子目录中生成“hybrid_numeric_Geno.txt”文件。

[0063]

```
（4） ### 输出数值化基因型
（5） output_numeric_geno <- TRUE
（6） if (output_numeric_geno==TRUE){
（7）   myGenoNewName <-
      paste0(gsub(".txt|.hmp","",basename(myGenoName)),"_numeric_Ge
      no.txt")
（8）   GD_DN2 <- as.data.frame(GD_DN)
（9）
（10）  if (!names(GD_DN2)[1]=="Markers"){
（11）    GD_DN2 <- cbind(row.names(GD_DN2),GD_DN2)
（12）    names(GD_DN2)[1] <- "Markers"
（13）  }
（14）
      write.table(GD_DN2,myGenoNewName,sep="\t",quote=F,row.name
      s=F)
（15）  rm(GD_DN2)
（16）}
```

[0064] 实施例2

[0065] 依据父本为自交系、母本为杂交种的三交种,利用生物信息学技术虚拟合成杂交种F1代基因型的方法,包括以下步骤:

[0066] 除了实施例1中的步骤(3)的(b)步骤外,其余步骤均同于实施例1。在步骤(3)的(b)步骤中,表型输入文件需要用@把合成母本的基因型分开,虚拟合成时,先合成母本的杂合基因型,再合成与父本杂交的杂合基因型。实施例1中的命令已经包含该过程。

[0067] 实施例3

[0068] 依据父本为杂交种、母本为杂交种的双交种基因型,利用生物信息学技术虚拟合成杂交种F1代基因型的方法,包括以下步骤:

[0069] 除了实施例1中的步骤(3)的(b)步骤外，其余步骤均同于实施例1。在步骤(3)的(b)中，表型输入文件需要用@把杂交成母本和父本的基因型分开，虚拟合成时，先分别合成母本和父本的杂合基因型，再合成最后的杂合基因型。实施例1中的命令已经包含该过程。

[0070] 上述实施例只是用于对本发明的举例和说明，而非意在将本发明限制于所描述的实施例范围内。此外本领域技术人员可以理解的是，本发明不局限于上述实施例，根据本发明的加倍虚拟拟合原理还可以做出更多种的变型和修改，这些变型和修改均落在本发明所要求保护的范围内。

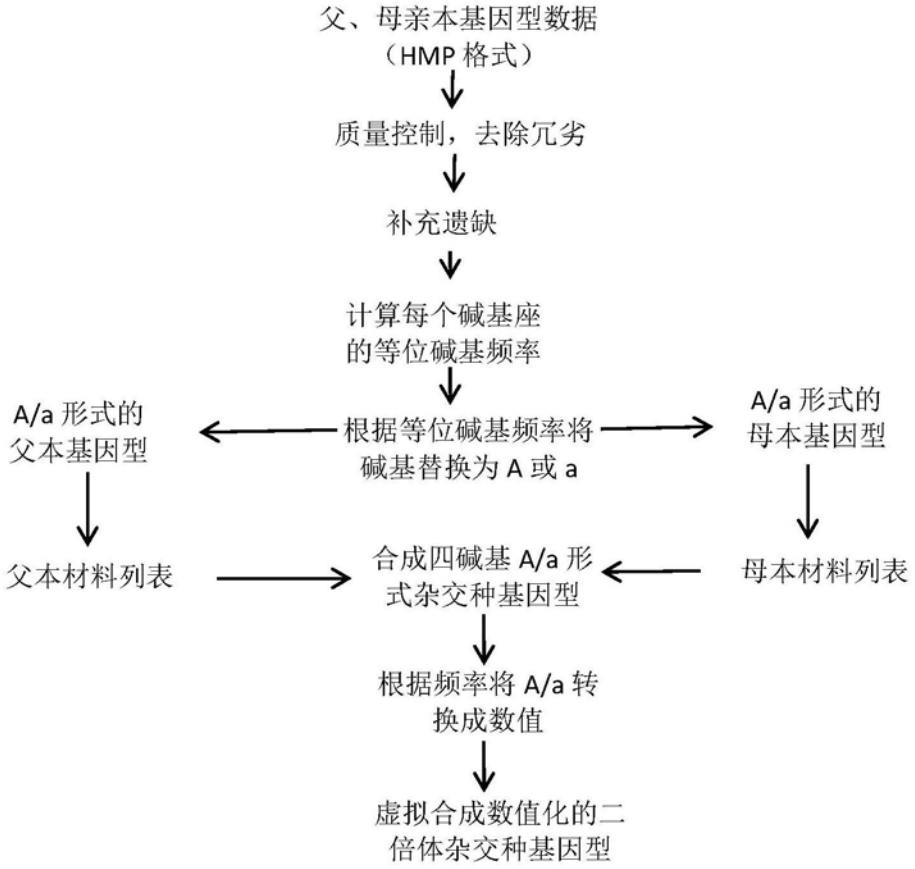
父、母亲本基因型数据
（HMP 格式）
质量控制，去除冗劣
补充遗缺
计算每个碱基座
的等位碱基频率
A/a 形式的
父本基因型
根据等位碱基频率将
碱基替换为 A 或 a
A/a 形式的
母本基因型
父本材料列表
合成四碱基 A/a 形
式杂交种基因型
母本材料列表
根据频率将 A/a 转
换成数值
虚拟合成数值化的二
倍体杂交种基因型

图1