个体单体型问题参数化算法研究

谢民主[1),2)] 陈建二[1)] 王建新[1)]

[1)](中南大学信息科学与工程学院 长沙 410083)

[2)](湖南师范大学物理与信息科学学院 长沙 410081)

摘 要 个体单体型问题指如何利用个体 DNA 测序片断数据,根据不同的优化准则确定该个体单体型的计算问题.因为技术上的限制,DNA 测序实验中能直接测定的片断长度是有限的,一个片断所覆盖的最大 SNP 位点数 k_1 通常小于 10;出于时间和金钱的考虑,覆盖一个 SNP 位点的最大片断数 k_2 也不是很大,通常约为 10 左右;与要测定的单体型 SNP 位点总数 n 及所测序的 DNA 片断总数 m 相比,k_1 和 k_2 均很小.在此基础上,文中对个体单体型问题最少 SNP 位点删除 MSR 和最少片段删除 MFR 模型进行了参数化,提出了时间复杂度分别为 $O(nk_1k_2+m\log m+mk_1)$ 和 $O(mk_2^2+mk_1k_2+m\log m+nk_2)$ 求解无空隙 MSR 和 MFR 的精确算法.和 Bafna 等提出的时间复杂度为 $O(mn^2)$ 和 $O(m^2n+m^3)$ 的精确算法相比,文中的算法效率提高了很多,具有较高的实用价值.

关键词 单核苷酸多态性;单体型;参数化算法;最少 SNP 位点删除;最少片断删除

中图法分类号 TP301 **DOI 号**: 10.3724/SP.J.1016.2009.01637

Research on Parameterized Algorithms of the Individual Haplotyping Problem

XIE Min-Zhu[1),2)] CHEN Jian-Er[1)] WANG Jian-Xin[1)]

[1)](*School of Information Science and Engineering, Central South University, Changsha* 410083)

[2)](*College of Physics and Information Science, Hunan Normal University, Changsha* 410081)

Abstract The Individual Haplotyping Problem is the computing problem of inducing an individual's haplotypes based on several optimal criteria from one's DNA sequence fragment data. Limited by current sequencing techniques, the maximum length of a fragment sequenced directly by DNA sequencers is not large, and the maximum number k_1 of SNP sites covered by a fragment is usually smaller than 10. In order to save money and time, the maximum number k_2 of fragments covering a SNP site is also small and about 10. Compared with the number n of SNPs of interest and the number m of fragments, k_1 and k_2 are both very small. Based on the above fact, this paper parameterizes MSR(Minimum SNP Removal) and MFR(Minimum Fragment Removal) models of the individual haplotyping problem, and introduces the corresponding exact algorithms to solve the gapless MSR and MFR problems in the time complexity $O(nk_1k_2+m\log m+mk_1)$ and $O(mk_2^2+mk_1k_2+m\log m+nk_2)$ respectively. Compared with Bafna et al.'s algorithms of time complexity $O(mn^2)$ and $O(m^2n+m^3)$ respectively, the algorithms are more efficient and applicable.

Keywords SNPs (Single-Nucleotide Polymorphisms); haplotype; parameterized algorithm; Minimum SNP Removal (MSR); Minimum Fragment Removal (MFR)

收稿日期:2007-07-30;最终修改稿收到日期:2009-07-06.本课题得到国家自然科学基金(60773111)、国家"九七三"重点基础研究发展规划前期研究专项基金(2008CB317107)、长江学者和创新团队发展计划(IRT0661)、湖南省自然科学基金(09JJ3116)、中国博士后科学基金及中南大学博士后科学基金资助.**谢民主**,男,1969 年生,博士,副教授,主要研究方向为生物信息学、精确和参数化算法.E-mail:xieminzhu@sina.com. **陈建二**,男,1954 年生,博士,教授,博士生导师,主要研究领域为生物信息学、精确和参数化算法、计算图形学和计算机网络. **王建新**,男,1969 年生,博士,教授,博士生导师,主要研究领域为计算机网络、计算机优化算法和生物信息学.

1 引 言

为了破译人类的遗传信息，1990 年 10 月人类基因组计划启动，2000 年 6 月人类基因组草图绘成，2003 年 4 月人类基因组图谱基本完成，至此人类基因组共性的一面被揭示出来，但人类个体多态性的一面没有得以体现. 不同的人具有不同的外貌、体格，对疾病有不同抵抗能力，对药物有不同的敏感性，从遗传上说，这是因为不同个体(除了同卵双胞胎外)的基因组不完全相同. 两个人之间的 DNA 差异约占基因组的 0.1%，单核苷酸多态性 SNPs(Single-Nucleotide Polymorphisms)为人类染色体某个位点上的碱基变化，SNPs 广泛分布在人类基因组中，在整个人类基因组中估计有几百万个 SNPs[1].

单核苷酸多态性是一个物种中不同个体表型的主要遗传来源. 识别 SNPs 对基因的精确定位、了解基因功能很有帮助，对遗传病等疾病的诊断和药物研究有重要作用，SNPs 可用于个体识别、亲子鉴定，亦可用于人类各群体的遗传关系分析. Stephens 等利用单体型研究人类 313 个基因中的 3899 个 SNPs，然后进行连锁不平衡分析，其结果支持了人类群体在近代扩张的说法[2]. Horikawa 等[3]根据 SNPs 进行关联分析在墨西哥裔美国人中把 2 型糖尿病基因定位在 2 号染色体长臂，并发现 CAPN10 基因的 3 个 SNPs 和 2 型糖尿病相关.

一个 SNP 位点指的是在一个物种的基因组 DNA 序列中不同个体可能出现不同碱基的位置. 在一条染色体 SNP 位点上的碱基序列叫做单体型(haplotype). 人类等二倍体生物的染色体是成对存在的，都有一对单体型. 图 1 中的一对单体型分别是“ATACG”和“GCATG”.

A . . T . . A . . C . . G
G . . C . . A . . T . . G

图 1 单体型

单体型在 SNPs 的上述应用中扮演着重要的角色，不幸的是在当前的实验技术下，把个体的一对染色体分开，独立测定每一条染色体上的单体型既费钱又费时间，因此利用计算机技术来确定个体的单体型具有极其重要的现实意义.

本文研究如何利用个体的 DNA 测序片断数据，根据不同的优化准则确定该个体的单体型. 本文第 2 节介绍个体单体型问题的抽象模型和当前的研究现状；第 3 节阐述基于无空隙片断数据的参数化算法，分析其复杂度；第 4 节给出实验结果；最后进行总结和展望.

2 个体单体型问题(Individual Haplotyping Problem)及当前研究现状

人类等双倍体生物的 DNA 序列是按染色体成对出现的. 对于任意一个 SNP 位点来说，2 条染色体上的碱基可以是相同的，这种现象叫纯合(homozygous)；也可以是不同的，这种现象叫做杂合(heterozygous). 这样一条染色体在 SNP 位点的投影序列即单体型(haplotype)可以用字符集{A,B}上的字符序列来表示，不必用真正的碱基字符，其中‘A’通常表示人群在该位点上常见的 SNP 值，这样图 1 中的一对单体型则可表示为一对字符串“ABAAB”和“BAABB”.

基于直接测定单体型的技术困难性和单体型在遗传分析上的重要性，Lancia 等[4]提出了直接利用个体的 DNA 测序片断数据来确定该个体的一对单体型，这个计算问题就是个体单体型问题(individual haplotyping problem).

对基因组进行测序时，因为技术上的限制，只能直接对较短的 DNA 片断进行测序，这些片断来自一对染色体的不同单体，测序过程中不可避免地会发生一些错误. 给定某个个体一组已测序的 DNA 片断数据，个体单体型问题就是要去掉最小数量的数据[4]，以便能发现一对单体型与剩下的数据兼容. 去掉最小数量的数据可以是去掉最少的片断或最少的 SNP 位点.

n 个 SNP 位点按在染色体上的次序从左到右记作 $S:\{1,2,\cdots,n\}$，m 个片断记作 $F:\{1,2,\cdots,m\}$. 任意 SNP 位点应该被某些 DNA 片断覆盖，任意片断在它所覆盖的 SNP 位点的取值为{A,B,-}，其中‘-’为空值，表示片断在该位点的取值未知. DNA 片断的数据集可以表示为在{A,B,-}上的一个 $m\times n$ 的矩阵，叫做 SNP 矩阵 $\boldsymbol{M}$[4]. 图 2 是一个 8×8 的 SNP 矩阵. SNP 矩阵的列表示 SNP 位点，行表示片段在对应的 SNP 位点上的取值，$\boldsymbol{M}_{ij}$ 表示第 i 个片断在第 j 个 SNP 位点上的取值.

SNP 位点

片段								
	-	-	-	-	-	B	A	B
	-	B	A	A	-	A	B	-
	A	B	A	A	-	-	-	-
	B	A	B	B	A	B	-	-
	-	A	B	B	-	-	-	-
	-	-	A	B	-	-	-	-
	-	-	-	A	B	A	B	A
	-	-	-	-	-	B	A	B

图 2　SNP 矩阵

为了行文简洁,下面引入与 SNP 矩阵 $\boldsymbol{M}$ 相关的几个定义.

定义 1.　如果 $(\exists k(M_{ik}\neq\text{'-'})\wedge(k\leq j))\wedge(\exists r(M_{ir}\neq\text{'-'})\wedge(j\leq r))$,则称行 i 覆盖列 j.

行 i 覆盖列 j 就是行 i 在列 j 上取值非空,或在列 j 前至少有一列,在列 j 后也至少有一列,使得行 i 在这两列上的取值均非空. 在图 2 中行 2 覆盖列 2～7. 如果某一行覆盖某一列,但是该行在该列上的取值为空,则称该行在该列上有个洞(hole). 图 2 中行 2 在列 5 上有个洞. 如果 $\boldsymbol{M}$ 的所有行都没有洞,则称 $\boldsymbol{M}$ 为无空隙的 SNP 矩阵.

定义 2.　如果两行在某一列上的值都不是空值,且这两行在该列上的值不相等,那么这两行在该列上冲突. 如果两行在所有的列上均不冲突,则这两行兼容.

如果测序过程没有任何错误,代表来自于同一染色体的片段的行必定两两兼容.

定义 3.　对于 $\boldsymbol{M}$ 的两列 j_1 和 j_2,如果 $\boldsymbol{M}$ 的所有行都可以划分到两个子集中,使得划分在同一个子集中的任意两行在列 j_1 和 j_2 上均不冲突,则列 j_1 和 j_2 兼容,满足上述条件的划分叫做在列 j_1 和 j_2 上兼容的划分;如果不存在着在列 j_1 和 j_2 上兼容的划分,则列 j_1 和 j_2 冲突.

在图 2 中,如果要在列 3 上不冲突,行 2,3,6 必须划分到同一子集中,行 4,5 则必须划分到另一子集中;但是这样划分的话,行 6 与行 2 在列 4 上冲突,由此可见,列 3 和 4 冲突. 对于列 1 和列 2 而言,把行 2 和 3 分到一个子集中,行 4 和 5 分到另一个子集中的任意划分都是在列 1 和 2 上兼容的划分,所以列 1 和列 2 兼容.

定理 1.　对于一个 SNP 矩阵 $\boldsymbol{M}$,只有'A'或'B'值的任意列不会和其它列冲突. 对于既有'A'又有'B'值的列 j_1, j_2,列 j_1, j_2 冲突当且仅当存在着两行 i_1、i_2:$i_1\neq i_2$,有 $M_{i_1j_1}$,$M_{i_1j_2}$, $M_{i_2j_1}$, $M_{i_2j_2}$ 均不为空值,且这 4 个值之中有 3 个相同、1 个不同.

证明.　不失一般性,令列 j_1 只有'A'值,对其它任意列 j_2,$\boldsymbol{M}$ 中的所有行均可如下划分:在列 j_2 上取 A 值的行划分到第一个集合中,在列 j_2 上取 B 值的行划分到第 2 个集合中,其它的行随机划分到这两个集合中,容易验证,这样划分到同一个集合中的任意两行在列 j_1 和 j_2 上不会冲突. 因此如果一列只有'A'或'B'值,该列不会和其它任何列冲突.

对于既有'A'又有'B'值的列 j,如果要使划分在同一个子集中的行在列 j 上不冲突,则在列 j 上取值为'A'的所有行必须划分在同一子集中,在列 j 上取值为'B'的行必须划分在另一个子集中. 对于既有'A'又有'B'值的两列 j_1, j_2, 如果存在着两行 i_1, i_2, 有 $M_{i_1j_1}$, $M_{i_1j_2}$, $M_{i_2j_1}$, $M_{i_2j_2}$ 均不为空值,且这 4 个值之中有 3 个相同、1 个不同,不失一般性,令 $M_{i_1j_1}=M_{i_2j_1}=M_{i_1j_2}=$'A'、$M_{i_2j_2}=$'B',那么按列 j_1 上的取值,行 i_1 和 i_2 应该划分到同一个子集中;而按列 j_2 上的取值,行 i_1 和 i_2 应该划分到不同的子集中,这相互矛盾,说明不存在列 j_1 和 j_2 上兼容的划分. 因此列 j_1, j_2 冲突.

反过来,对于既有'A'又有'B'值的两列 j_1, j_2,如果列 j_1, j_2 不冲突,那么存在着一个划分使 $\boldsymbol{M}$ 的所有行划分到两个子集中,使得相同子集中的任意两行在列 j_1 和 j_2 上不冲突. 这样对任意两行 i_1, i_2, 如果有 $M_{i_1j_1}$, $M_{i_1j_2}$, $M_{i_2j_1}$, $M_{i_2j_2}$ 均不为空值,即不是'A'值就是'B'值,那么如果这两行被划分到同一个子集中,则这两行在列 j_1 上的值相等,在列 j_2 上的值也应相等;如果这两行被划分到不同的子集中,则这两行在列 j_1 上的值不相等,在列 j_2 上的值也应不相等. 这样这 4 个值之中就不可能有 3 个相同、1 个不同的情况. 定理得证.　　证毕.

定义 4.　如果 SNP 矩阵 $\boldsymbol{M}$ 的所有行可以分成 2 个不相交的子集,每个子集中的所有行都相互兼容,则 $\boldsymbol{M}$ 是可行的.

如果 $\boldsymbol{M}$ 是可行的,则很容易找到一个划分,使划分在同一个子集中的所有行都兼容,进而通过同一个子集中的行很容易重建与这些行兼容的单体型. 如果测序过程没有任何错误,测序片段数据对应的 SNP 矩阵必定是可行的,那么通过 DNA 测序片段数据很容易得出个体的一对单体型. 可是由于测序过程的错误是不可避免的,因此实验室测得的片段数据对应的 SNP 矩阵通常是不可行的.

Lancia 等[4] 最先讨论对 SNP 矩阵进行处理使其可行的计算模型,引入了下面的个体单体型优化问题:

问题 1. MFR(Minimum Fragment Removal,最少片断删除). 给定一个 SNP 矩阵 $\boldsymbol{M}$,删除最少的行使 $\boldsymbol{M}$ 可行.

问题 2. MSR(Minimum SNP Removal,最少 SNP 删除). 给定一个 SNP 矩阵 $\boldsymbol{M}$,删除最少的列使 $\boldsymbol{M}$ 可行.

不进行任何处理,图 2 所示的 SNP 矩阵是不可行的. 当去掉第 6 行后,行 1,4,5 和 8 相互兼容,构成一个子集,进而得出一个单体型为"BABBA-BAB";行 2,3 和 7 相互兼容,构成另一个子集,进而得出另一个单体型为"ABAABABA". 或者不去掉行,则去掉列 4 后,行 1,4,5 和 8 相互兼容,构成一个子集,进而得出一个单体型为"BAB-ABAB";行 2,3,6 和 7 相互兼容,构成另一个子集,进而得出另一个单体型为"ABA-BABA"(去掉的列所代表 SNP 位点的值为空值).

Lancia 等[4] 证明了在 DNA 片断数据有洞(hole)的情况下,MFR 是 NP-hard;如果片断上洞的个数超过一个,则 MSR 是 NP-hard;并且在这些情况下,MSR 和 MFR 问题都是 APX-hard. 在 DNA 片断数据中无洞的情况下,Lancia 等[4] 证明了 MFR 和 MSR 是多项式时间可解的. Bafna 等[5] 对这些问题进行了深入的研究,对于 m 个 DNA 片断,n 个 SNP 位点的无空隙的 SNP 矩阵的 MSR 和 MFR 问题,提出了时间复杂度为 $O(mn^2)$ 和 $O(m^2n+m^3)$、空间复杂度为 $O(mn+n^2)$ 和 $O(mn+m^3)$ 的多项式算法.

人有 24 条不同的染色体,染色体的平均长度约为 150000000 个碱基①,SNP 的分布密度约为 0.1%[6-7],这样一条染色体上的 SNP 位点数约为 150000. 当从 shotgun 全基因组测序所得到的序列数据推出个体的单体型时,能直接测序的片段长度约为 1000 个碱基,一个碱基大约被 10 个不同的片段覆盖,因此得到的片断数约为 1500000(150000,000×10/1000). 在这种情况下,即使 DNA 片断数据无空隙,解决 MSR 和 MFR 的这些算法的时间复杂度仍然太大,推断染色体一个区域的单体型时情况也是如此. 因而根据目前基因组测序的技术现状,针对 DNA 测序数据和个体单体型问题的特点提出更高效的算法具有重要的现实意义.

3 无空隙片断数据的参数化算法

目前最流行的 SNP 探测方法是 DNA 直接测序[8-9],这种方法 48h 可分析近百万个碱基对,杂合的 SNPs 探测率达 95%以上,SNP 联盟(SNP Consortium)用这种方法测得了上百万 SNPs[6,9]. 当前 DNA 测序的主导方法是 Sanger 双脱氧链终止法[10]. 采用 Sanger 双脱氧链终止法测序,一次能测定的 DNA 序列的长度仅为 800～1200 个碱基. 各大测序中心使用的第三代测序仪如 MageBACE 等可测片断长度约为 1000 碱基②,而 SNPs 的平均分布密度约为 1/1000[6-7],虽然 SNPs 在整个染色体上的分布很不均匀,从已有的数据来看,一个长度 1000bp 的片断上的 SNP 位点是极其有限的,通常在 10 个以内[11-12].

另外出于测序的时间和代价的考虑,基因组测序的 DNA 片断对 SNP 位点的覆盖度(覆盖一个 SNP 位点的片段数)是有限的. 在目前的 shotgun 实验测序中,覆盖度约为 10[13]. 当需要测定某个体在一条染色体上的单体型时,覆盖一个 SNP 位点的 DNA 片断数远小于 shotgun 法测得的片断总数. 因此,在 DNA 测序实验中,一个片断覆盖的 SNP 位点数(小于 10)和覆盖一个 SNP 位点的片断数(约为 10)与通常要探测的 SNP 位点数(n)及 DNA 片断总数(m)相比,均是很小的数.

基于以上事实,我们提出以下参数化条件.

定义 5. (k_1, k_2)参数化条件:k_1, k_2 是正整数,(k_1, k_2)参数化条件定义为片段覆盖的 SNP 位点数不超过 k_1,覆盖任意 SNP 位点的片段数不超过 k_2. 对一个无空隙的 SNP 矩阵 $\boldsymbol{M}$ 而言,(k_1, k_2)参数化条件等价为矩阵 $\boldsymbol{M}$ 的每一行最多只有一块连续非'-'字符序列,该字符序列长度不超过 k_1 个;矩阵 $\boldsymbol{M}$ 每一列最多有 k_2 个行的取值为非'-'字符.

下面本文深入研究无空隙的 SNP 矩阵 $\boldsymbol{M}$ 的 MSR 和 MFR 算法. 对于任意无空隙的 SNP 矩阵 $\boldsymbol{M}$,很容易得出它的(k_1, k_2)参数值,因此为了行文简洁,除了特别声明外,以下的 SNP 矩阵都是指满足(k_1, k_2)参数化条件的无空隙 SNP 矩阵.

首先对 MSR 问题和 MFR 问题参数化.

问题 3. P_MSR(Parameterized Minimum SNP Removal). 给定一个 SNP 矩阵 $\boldsymbol{M}$,满足(k_1, k_2)参数化条件,P_MSR 问题是要求删除最少的列,也就是保留最多的列使 $\boldsymbol{M}$ 可行.

① http://www.ornl.gov/sci/techresources/Human_Genome/publicat/primer/prim1.html#3

② http://www.ebiotrade.com/custom/amersham/MegaBACE.htm

问题 4.　P_MFR(Parameterized Minimum Fragment Removal). 给定一个 SNP 矩阵 $\boldsymbol{M}$,满足 (k_1, k_2)参数化条件,P_MFR 问题是要求删除最少的行,也就是保留最多的行使 $\boldsymbol{M}$ 可行.

本文中,一个 SNP 矩阵 $\boldsymbol{M}$ 的 P_MSR 解指的是使 $\boldsymbol{M}$ 可行能保留下来的最多的列数,用 P_MSR($\boldsymbol{M}$)来表示;一个 SNP 矩阵 $\boldsymbol{M}$ 的 P_MFR 解指的是使 $\boldsymbol{M}$ 可行能保留下的最多的行数,用 P_MFR($\boldsymbol{M}$)来表示.

函数 $left$ 和 $right$ 分别用来表示 $\boldsymbol{M}$ 中的行覆盖的最左边和最右边的列号,即行 i 覆盖的最左边的列是 $left(i)$,覆盖的最右边的列是 $right(i)$.$\boldsymbol{M}$ 的前 i 行构成的 SNP 矩阵记作 $\boldsymbol{M}(i,:)$.

在求解问题 3 和问题 4 时,先对 SNP 矩阵 $\boldsymbol{M}$ 进行如下预处理:

1. 排序. 调整 $\boldsymbol{M}$ 中各行的次序,使各行按其 $left$ 值进行非降序排列. 同时对于任意列 j,计算出覆盖该列的行的序号的有序集,记作 $rowset(j)$.

对于一个满足(k_1, k_2)参数化条件的 SNP 矩阵 $\boldsymbol{M}$, 排序后如图 3 所示.

2. 去掉冗余列. 对 $\boldsymbol{M}$ 的每一列 j,如果 $rowset(j)$中的所有行在该列的取值中没有出现'A'或没有出现'B',那么该列就是冗余列,标记该列. 最后去掉所有标记的列,调整剩下列的序号,修改对应行的 $left$ 和 $right$ 函数值.

3. 去掉冗余行:对 $\boldsymbol{M}$ 的每一行 i,如果该行在剩下的列上的取值均为空,则该列就是冗余行,标记该行. 最后去掉所有标记的行,修改受影响的列的 $rowset$ 集.

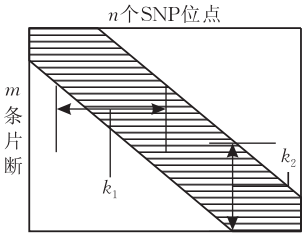


图 3　预处理后的 SNP 矩阵示意图

令 $\boldsymbol{M}$ 采用如下数据结构:对每一行 i,记下该行覆盖的最左边和最右边的列号,即 $left(i)$ 和 $right(i)$,记录该行从列 $left(i)$到列 $right(i)$的各列上的值.

预处理时间复杂度分析:排序所需的时间为 $O(m\log m)$,对排好序的行扫描一遍可得到各列的 $rowset$ 有序集,这样步 1 所需时间为 $O(m\log m+mk_1)$;去掉冗余列所需时间为 $O(nk_2)$,修改 $left$ 和 $right$ 值所需时间为 $O(mk_1)$,这样步 2 所需的时间为 $O(nk_2+mk_1)$;去掉冗余行所需时间为 $O(mk_1)$,修改 $rowset$ 有序集所需时间为 $O(nk_2)$,这样步 3 所需的时间为 $O(mk_1+nk_2)$. 所以整个预处理所需的时间为 $O(m\log m+nk_2+mk_1)$.

定理 2.　一个满足(k_1, k_2)参数化条件的 SNP 矩阵 $\boldsymbol{M}$ 经过以上预处理后仍然满足(k_1, k_2)参数化条件.

证明.　排序、去掉冗余列和冗余行的预处理显然不会增加某一行覆盖的列数和覆盖某一列的行数.　　证毕.

定理 3.　一个 SNP 矩阵 $\boldsymbol{M}$ 经过以上预处理后得到的 SNP 矩阵记作 $\boldsymbol{M}'$,去掉的冗余列集和冗余行集分别记作 $\boldsymbol{X}$ 和 $\boldsymbol{Y}$,则 $\boldsymbol{M}$ 的 P_MSR 解等于 $\boldsymbol{M}'$的 P_MSR 解与 $\boldsymbol{X}$ 的列数之和,$\boldsymbol{M}$ 的 P_MFR 解等于 $\boldsymbol{M}'$的对应解与 $\boldsymbol{Y}$ 的行数之和.

证明.　首先证明 $\boldsymbol{M}$ 的 P_MSR 解等于 $\boldsymbol{M}'$的 P_MSR解与 $\boldsymbol{X}$ 的列数之和. 因为去掉的列在所有的行上的取值只含有'A'或'B',所以 $\boldsymbol{M}$ 的任意一行均不会在去掉的列上和其它行冲突. 令 $\boldsymbol{S}$ 是 $\boldsymbol{M}'$的一些或全部列构成的集合,容易看出,如果保留 $\boldsymbol{S}$ 中的所有列可使 $\boldsymbol{M}'$可行,那么保留 $\boldsymbol{S}\cup\boldsymbol{X}$ 中的所有列也可使 $\boldsymbol{M}$ 可行. 由此可证 $\boldsymbol{M}$ 的 P_MSR 解等于 $\boldsymbol{M}'$的 P_MSR 解与 $\boldsymbol{X}$ 的列数之和.

同理 $\boldsymbol{Y}$ 中的行不会与其它任何行冲突,因此 $\boldsymbol{M}'$的 P_MFR 解中所保留下的行加上 $\boldsymbol{Y}$ 中的行所形成的 SNP 矩阵肯定是可行的,因此可以证明 $\boldsymbol{M}$ 的 P_MFR解等于 $\boldsymbol{M}'$的 P_MFR 解与 $\boldsymbol{Y}$ 的行数之和.　　证毕.

从定理 3 可知,求解一个 SNP 矩阵的 P_MSR 和 P_MFR 解可以归结为求其预处理后的 SNP 矩阵的对应解. 对于行列数为 0 的 SNP 矩阵,易知其 P_MSR 和 P_MFR 解均为 0. 为了叙述的简洁,下面讨论的 SNP 矩阵 $\boldsymbol{M}$ 均指预处理后的 SNP 矩阵,该矩阵具有如下特点:行列数 $m,n>0$;任何一个列总有一些行的值是'A',还有一些行的值为'B';对于行 $i\leq j$, 有 $left(i)\leq left(j)$.

3.1　P_MSR 算法

由文献[4]的定理 3 可知一个无空隙的 SNP 矩阵 $\boldsymbol{M}$ 是可行的当且仅当其任意两列均不冲突,因此 P_MSR 问题等价于保留最多的列,使保留的列中的任意两列均不冲突.

令 C 是 $\boldsymbol{M}$ 一个列的子集,如果 C 中的列彼此兼容,且 C 中的列的最大列号为 j,则 C 称为列 j 的兼容列集. 对于列 j 的任意兼容列集 C,容易看出保留 C 中的所有列可使 $\boldsymbol{M}$ 可行. 令 $K(j)$表示列 j 的最大兼容列集. 显然:

$$K(1)=\{1\} \tag{1}$$

由文献[4]的引理 2 可知对于一个无空隙的 SNP 矩阵 $\boldsymbol{M}$ 的任意 3 列 $j_1<j_2<j_3$，如果 j_1 与 j_2 兼容，j_2 与 j_3 兼容，则一定有 j_1 与 j_3 兼容. 因此对于任意两列 r、j：$r<j$，如果列 r 和 j 兼容，则 $K(r)\cup\{j\}$ 一定是列 j 的兼容列集，由此容易证明 $K(j)=\{j\}\cup\max_{r:1\le r<j,\text{列}r\text{和}j\text{兼容}}K(r)$，其中对集合进行的 max 运算是取元素最多的集合，即 $\max_{r:1\le r<j,\text{列}r\text{和}j\text{兼容}}K(r)$ 为满足 $1\le r<j$ 且列 r 和 j 兼容的所有 $K(r)$ 中的最大集合，如果没有满足这些条件的 $K(r)$，则为空集 $\varnothing$，下文也是如此.

对于列 j 和它前面的列 r，下面判断列 j 是否和 r 相容：

Case 1. $r\le j-k_1$：因为 $\boldsymbol{M}$ 满足 (k_1,k_2) 参数化条件，所以 $\boldsymbol{M}$ 的任一行覆盖的列数不会超过 k_1，这样就不可能存在着两行 i_1、i_2：$i_1\ne i_2$，$M_{i_1j_1}$，$M_{i_1j_2}$，$M_{i_2j_1}$，$M_{i_2j_2}$ 均不为空值. 根据定理 1，列 r 与 j 兼容.

Case 2. $r>j-k_1$：因为 $\boldsymbol{M}$ 经过了预处理，所以任意一列上既有‘A’又有‘B’. 根据定理 1，只需对在列 r 与 j 上取值均非空的行进行观察，这些行的个数不大于 k_2. 如果这些行在这两列上的取值只有“AA”、“BB”两种类型，或只有“AB”、“BA”两种类型，那么列 r 与 j 兼容；否则就不兼容.

令 $A(j)$ 表示 $K(1)$ 到 $K(j)$ 的最大值，如果 $j<0$，规定 $A(j)=\varnothing$；令 $OK(j)=\{r|j>r>j-k_1$ 且列 r 与 j 兼容$\}$. 由上面的讨论，易知下面的公式成立：

$$K(j)=\{j\}\cup\max(A(j-k_1),\max_{r\in OK(j)}(K(r))) \quad (2)$$

由 P_MSR 问题的定义可知：

$$\text{P\_MSR}(M)=\max(|A(n-k_1)|,|K(n-k_1+1)|,\cdots,|K(n)|) \quad (3)$$

在式(1)～(3)的基础上，可得到求解无空隙 SNP 矩阵 $\boldsymbol{M}$ 的 P_MSR 问题的动态规划算法，如图 4 所示.

```
P_MSR 算法
Input: 预处理后得到的无空隙 SNP 矩阵 M,令其行数为 m,列数为 n
Output: M 的 P_MSR 解
1. 扫描 M 得到一行覆盖的最大列数 k1 和覆盖一列的最大行数 k2,K(1)=A(1)={1};  //式(1)
2. for j=2…n do //根据式(2)递推
2.1.  if j≤k1 then Kp=∅;
      else Kp=A(j-k1);
2.2.  for r=max(1,j-k1)…j-1 do
2.2.1.  if 列 r 与 j 兼容 and |Kp|<|K(r)|
          then Kp=K(r);
2.3.  K(j)={j}∪Kp;
3. if n≤k1 then Kp=∅ else Kp=A(n-k1);
4. for r=max(1,n-k1)…n do  //式(3)
4.1.  if |Kp|<|K(r)| then Kp=K(r);
5. return |Kp|;
```

图 4 P_MSR 算法

定理 4. 对于一个 $m\times n$ 满足 (k_1,k_2) 参数化条件的无空隙 SNP 矩阵 $\boldsymbol{M}$，P_MSR 算法是正确的，加上预处理其时间复杂度为 $O(nk_1k_2+m\log m+mk_1)$，空间复杂度为 $O(mk_1+nk_1)$.

证明. P_MSR 算法的正确性由式(1)～(3)的正确性来保证，而式(1)～(3)的正确性在引入的时候已经得到了证明. 下面分析其复杂度.

时间复杂度分析. 步 1 中扫描 $\boldsymbol{M}$ 得到 k_1 和 k_2 所需的时间为 $O(mk_1)$；步 2.2.1 中判断列 r 与 j 是否兼容所需时间为 $O(k_2)$，步 2.2.1 最多执行 nk_1 次，由此可知步 2 的时间复杂度为 $O(nk_1k_2)$；步 3 的时间复杂度为 $O(1)$；步 4 的时间复杂度为 $O(k_1)$. 这样加上预处理的时间整个算法的时间复杂度为 $O(nk_1k_2+m\log m+mk_1)$.

空间复杂度分析. SNP 矩阵所需的空间为 $O(mk_1)$，计算 $K(j)$ 只需要 $A(j-k_1)$，$K(j-k_1+1)$，…，$K(j-1)$ 的值，因此 A 和 K 均可采用大小为 k_1 的循环队列，需要的空间为 $O(nk_1)$，所以整个算法的空间复杂度为 $O(mk_1+nk_1)$. 定理得证.

证毕.

3.2 P_MFR 算法

考虑 $m\times n$ 的 SNP 矩阵 $\boldsymbol{M}$ 的前 i 行构成的矩阵 $\boldsymbol{M}(i,:)$，令 R 为 $\boldsymbol{M}(i,:)$ 的行的子集，如果保留 R 中的所有行能使 $\boldsymbol{M}(i,:)$ 可行，那么 R 就称为 $\boldsymbol{M}(i,:)$ 的一个兼容行集.

如果 R 是 $\boldsymbol{M}(i,:)$ 的一个兼容行集，则必定可以把 R 划分成两个子集 R_1 和 R_2，使得划分在同一子集的行彼此兼容. 对于 R_j：$j=1,2$，找出具有以下特性的行 r_j 作其代表：(1) $r_j\in R_j$；(2) 对于任意行 $r\in R_j$，有 $right(r)<right(r_j)$，或者 $right(r)=right(r_j)$ 且 $r\le r_j$. 在“$right(r_1)<right(r_2)$”或“$right(r_1)=right(r_2)$ 且 $r_1<r_2$”的情况下，R 叫做行 r_1，r_2 代表的 $\boldsymbol{M}(i,:)$ 的兼容行集；否则 R 叫做行 r_2，r_1 代表的 $\boldsymbol{M}(i,:)$ 的兼容行集.

划分得来的子集在极端情况下可能是空集，为了上述概念的完整性，规定空集的代表行为 0，而且 $left(0)=-1$，$right(0)=-2$，行 0 与所有其它行均兼容. 显然空集也应是 $\boldsymbol{M}(i,:)$ 的一个兼容行集，所以进一步规定空集是行 0，0 代表的 $\boldsymbol{M}(i,:)$ 的兼容行集. 令 $R(d,k,i)$ 表示以行 d，k 为代表的 $\boldsymbol{M}(i,:)$ 的最大兼容行集. 易知

$$R(0,0,1)=\varnothing;\ R(0,1,1)=\{1\} \quad (4)$$

令 $R(*,k,i)$ 表示所有满足 $right(d)<left(i+1)$ 的最大 $R(d,k,i)$，即 $R(*,k,i)=$

$\max_{d;right(d)<left(i+1)} R(d,k,i)$. 令 $R(*,*,i)$ 表示所有满足 $right(k)<left(i+1)$ 的最大 $R(*,k,i)$,即 $R(*,*,i)=\max_{k;right(k)<left(i+1)} R(*,k,i)$. 这样就有

$$R(*,0,1)=\varnothing;R(*,1,1)=\{1\};$$

$$R(*,*,1)=\begin{cases}\{1\}, & right(1)<left(2)\\ \varnothing, & \text{否则}\end{cases} \tag{5}$$

为了使 $R(*,k,i)$、$R(*,*,i)$ 在 $i=m$ 时有意义,引入行 $m+1$,规定 $left(m+1)=n+1$. 这样,根据 P_MFR 的定义,显然有

$$\text{P\_MFR}(\boldsymbol{M})=|R(*,*,m)| \tag{6}$$

即 $R(*,*,m)$ 中的行数.

定理 5.　对于一个预处理后的无空隙 SNP 矩阵 $\boldsymbol{M}$,行 $i+1$ 与 $R(d,k,i)$ 中以行 d(或 k)为代表的子集中所有行兼容当且仅当行 $i+1$ 与行 d(或 k)兼容.

证明.　如果行 $i+1$ 与 $R(d,k,i)$ 中以行 d(或 k)为代表的子集中所有行兼容,那么显然行 $i+1$ 与行 d(或 k)兼容.

下面证明如果行 $i+1$ 与行 d 兼容,则行 $i+1$ 与 $R(d,k,i)$ 中以行 d 为代表的子集中所有行兼容:对于 $R(d,k,i)$ 中以行 d 为代表的子集中任意一行 r,必有 $right(r)\le right(d)$,且行 r 与 d 在列 $\max(left(r),left(d))$ 到列 $right(r)$ 中的任一列上取值必定非空(无空隙),而且相等. 由于 $\boldsymbol{M}$ 是经过排序的,且 $r,d\le i+1$,所以 $\max(left(r),left(d))\le left(i+1)$. 由于 $\boldsymbol{M}$ 是无空隙的 SNP 矩阵,且行 $i+1$ 与行 d 兼容,所以行 $i+1$ 与行 d 在列 $left(i+1)$ 到列 $\min(right(i+1),right(d))$ 中的任一列上取值必定相等,而且非空. 这样行 $i+1$ 与行 r 在列 $left(i+1)$ 到列 $\min(right(i+1),right(r))$ 中的任一列上取值必定相等,而在其它的列,总有一行在该列的值为空,因此行 $i+1$ 与行 r 在任意列上均不冲突,行 $i+1$ 与行 r 兼容.

同样可以证明如果行 $i+1$ 与行 k 兼容,则行 $i+1$ 与 $R(d,k,i)$ 中以行 k 为代表的子集中所有行兼容. 定理得证.　　证毕.

为了行文简洁,满足下列 2 个条件的行 k 的集合记作 $S_k(i)$:(1) $k\le i$; (2) $right(k)\ge left(i)$.

满足下列 4 个条件的 (d,k) 的集合记作 $S_{dk}(i)$:

(1) $d,k<i$;

(2) $right(d)\ge left(i)$;

(3) $right(k)\ge left(i)$;

(4) "$right(d)<right(k)$"或"$right(d)=right(k)$且$d<k$".

令 $C(i)=\{r\mid r\in S_k(i)$,且行 r 与 i 兼容$\}$.

下面讨论对于行 i: $2\le i\le m$,在 $R(*,*,i-1)$、所有的 $R(*,k,i-1):k\in S_k(i)$ 和所有的 $R(d,k,i-1):(d,k)\in S_{dk}(i)$ 都已知的条件下,如何求出 $R(*,*,i)$、所有可能的 $R(*,k,i)(k\in_k(i+1))$ 和所有的 $R(d,k,i)((d,k)\in S_{dk}(i+1))$.

首先对于所有的 $(d,k)\in S_{dk}(i)$,计算 $R(d,k,i)$:

$R(d,k,i-1)$ 显然是一个以行 d,k 为代表的 $\boldsymbol{M}(i,:)$ 的兼容行集,而 $\boldsymbol{M}(i,:)$ 比 $\boldsymbol{M}(i-1,:)$ 多的行只有行 i,易知 $R(d,k,i)$ 比 $R(d,k,i-1)$ 最多只会增加一个元素,增加的元素只可能是行 i. 在满足"i 与 d 兼容,$right(i)<right(d)$"或"i 与 k 兼容,$right(i)<right(k)$"的条件下,根据定理 5,$\{i\}\cup R(d,k,i-1)$ 显然是一个以行 d,k 为代表的 $\boldsymbol{M}(i,:)$ 的兼容行集,所以有

$$R(d,k,i)=\{i\}\cup R(d,k,i-1) \tag{7}$$

否则,必定有

$$R(d,k,i)=R(d,k,i-1) \tag{8}$$

这是因为如下事实:条件"i 与 d 兼容,$right(i)\le right(d)$"得不到满足,则行 i 如果划分到行 d 代表的子集中,那么或者 d 所在的子集中的行不再相互兼容,或者 d 所在的子集的代表不再是 d 而应该是 i. 而条件"i 与 k 兼容,$right(i)\le right(k)$"得不到满足,那么或者 k 所在的子集中的行不再相互兼容,或者 k 所在的子集的代表不再是 k 而应该是 i. 这两个条件都得不到满足,则行 i 必定不能在以 d,k 为代表的 $\boldsymbol{M}(i,:)$ 的兼容行集之中.

对于所有的 $r\in S_k(i)$:

如果 $right(r)\le right(i)$,有下面的公式成立(证明见定理 6):

$$R(r,i,i)=\{i\}\cup\max\Big(R(*,r,i-1),\ \max_{d;d\in C(i),(d,r)\in S_{dk}(i)} R(d,r,i-1),\ \max_{k;k\in C(i),(r,k)\in S_{dk}(i),right(k)\le right(i)} R(r,k,i-1)\Big) \tag{9}$$

否则,即 $right(r)>right(i)$,有下面的公式成立(证明见定理 6):

$$R(i,r,i)=\{i\}\cup\max\Big(R(*,r,i-1),\ \max_{d;d\in C(i),(d,r)\in S_{dk}(i),right(d)\le right(i)} R(d,r,i-1)\Big) \tag{10}$$

对于所有的 $k\in S_k(i)$,计算 $R(*,k,i)$:

令 I 表示 $\max_{d;right(d)<left(i)} R(d,k,i)$,$I_{dk}(i)$ 表示

$S_{dk}(i)\bigcup\{(i,k)\mid k\in S_k(i),right(k)>right(i)\}$.

根据 $R(*,k,i)$ 的定义：

$$R(*,k,i)=\max(I,\max_{d;left(i)\le right(d)<left(i+1)}R(d,k,i))$$
$$=\max(I,\max_{d;(d,k)\in I_{dk}(i),right(d)<left(i+1)}R(d,k,i)).$$

对于所有的 $d\le left(i)$，如果 k 与 i 兼容且 $right(k)>right(i)$，有

$R(d,k,i)=R(d,k,i-1)\bigcup\{i\}$（理由与式(7)相同），

$$I=\max_{d;right(d)<left(i)}R(d,k,i-1)\bigcup\{i\}$$
$$=\{i\}\bigcup R(*,k,i-1);$$

否则有

$R(d,k,i)=R(d,k,i-1)$（理由与式(8)相同），

$$I=\max_{d;right(d)<left(i)}R(d,k,i-1)=R(*,k,i-1).$$

因此有下面公式成立：

如果 k 与 i 兼容且 $right(k)>right(i)$：

$$R(*,k,i)=\max(R(*,k,i-1)\bigcup\{i\},\max_{d;(d,k)\in I_{dk}(i),right(d)<left(i+1)}R(d,k,i)) \tag{11}$$

否则，必定有

$$R(*,k,i)=\max(R(*,k,i-1),\max_{d;(d,k)\in I_{dk}(i),right(d)<left(i+1)}R(d,k,i)) \tag{12}$$

最后计算 $R(*,i,i)$ 和 $R(*,*,i)$（证明见定理 6）：

$$R(*,i,i)=\max(\{i\}\bigcup R(*,*,i-1),\{i\}\bigcup\max_{k;k\in C(i),right(k)\le right(i)}R(*,k,i-1),\max_{d;d\in S_k(i),right(d)\le right(i),right(d)<left(i+1)}R(d,i,i)) \tag{13}$$

$$R(*,*,i)=\max(R(*,*,i-1),\max_{k;k\in S_k(i)\bigcup\{i\},right(k)<left(i+1)}R(*,k,i)) \tag{14}$$

由于 $S_k(i+1)$ 与 $S_k(i)$ 的差集或者是空或者是 $\{i\}$，$S_{dk}(i+1)$ 与 $S_{dk}(i)$ 的差集最多是 $\{(d,i)\mid d\in S_k(i),right(d)\le right(i)\}\bigcup\{(i,k)\mid k\in S_k(i),right(i)\le right(k)\}$，所以通过式(7)～(14)可以求出 $R(*,*,i)$、所有可能的 $R(*,k,i)(k\in S_k(i+1))$ 和所有的 $R(d,k,i)((d,k)\in S_{dk}(i+1))$.

根据式(4)～(14)，可得出下面的 P_MFR 动态规划算法，如图 5 所示.

P_MFR 算法
Input：预处理后的无空隙 SNP 矩阵 $\boldsymbol{M}$，令其行数为 m，列数为 n
Output：$\boldsymbol{M}$ 的 P_MFR 的解
1. $R(0,0,1)=\varnothing$，$R(0,1,1)=\{1\}$，$R(*,0,1)=\varnothing$；$R(*,1,1)=\{1\}$；　//式(4)，(5)
2. if $right(1)<left(2)$ then $R(*,*,1)=\{1\}$; else $R(*,*,1)=\varnothing$; //式(5)
3. for $i=2\cdots m$ do
3.1. for each $k\in S_k(i)$ do
3.1.1. if $right(k)\le right(i)$ then $R(k,i,i)=\{i\}\bigcup R(*,k,i-1)$; //用式(9)初始化
else $R(i,k,i)=\{i\}\bigcup R(*,k,i-1)$; //用式(10)初始化
3.1.2. $R(*,k,i)=R(*,k,i-1)$; //用式(12)初始化
3.1.3. if 行 k 与 i 兼容 then $C(k)=1$; else $C(k)=0$;
3.2. $R(*,i,i)=\{i\}\bigcup R(*,*,i-1)$，$R(*,*,i)=R(*,*,i-1)$; //用式(13)、(14)初始化
3.3. for each $(d,k)\in S_{dk}(i)$ do
3.3.1. if $(C(d)=1,right(i)<right(d))$ or $(C(k)=1,right(i)<right(k))$ then $R(d,k,i)=\{i\}\bigcup R(d,k,i-1)$; //式(7)
else $R(d,k,i)=R(d,k,i-1)$; //式(8)
3.3.2. if $C(d)=1,right(k)\le right(i)$ then $R(k,i,i)=\max(R(k,i,i),\{i\}\bigcup R(d,k,i-1))$; //式(9)
3.3.3. if $C(k)=1,right(k)\le right(i)$ then $R(d,i,i)=\max(R(d,i,i),\{i\}\bigcup R(d,k,i-1))$; //式(9)
3.3.4. if $C(d)=1,right(d)\le right(i)<right(k)$ then $R(i,k,i)=\max(R(i,k,i),\{i\}\bigcup R(d,k,i-1))$; //式(10)
3.4. for each $k\in S_k(i)$ do
3.4.1. if $C(k)=1,right(k)>right(i)$ then $R(*,k,i)=\{i\}\bigcup R(*,k,i-1)$; //式(11)
3.4.2. if $C(k)=1,right(k)\le right(i)$ then $R(*,i,i)=\max(R(*,i,i),\{i\}\bigcup R(*,k,i-1))$; //式(13)
3.5. for each $(d,k)\in S_{dk}(i)$ do
3.5.1. if $right(d)<left(i+1)$ then $R(*,k,i)=\max(R(*,k,i),R(d,k,i))$; //式(11)，(12)
3.6. for each $r\in S_k(i)$ do
3.6.1. if $right(i)<right(r)$，$right(i)<left(i+1)$ then
$R(*,r,i)=\max(R(*,r,i),R(i,r,i))$; //式(11)，(12)
3.6.2. if $right(r)\le right(i)$，$right(r)<left(i+1)$ then $R(*,i,i)=\max(R(*,i,i),R(r,i,i))$; //式(13)
3.7. for each $k\in S_k(i)\bigcup\{i\}$ do
3.7.1. if $right(k)<left(i+1)$ then $R(*,*,i)=\max(R(*,*,i),R(*,k,i))$; //式(14)
4. return $|R(*,*,m)|$; //式(6)

图 5 P_MFR 算法

定理 6. 对于一个预处理后满足 (k_1,k_2) 参数化条件的无空隙 $m\times n$ SNP 矩阵 $\boldsymbol{M}$，P_MFR 算法是正确的，加上预处理，其时间复杂度为 $O(mk_2^2+mk_1k_2+m\log m+nk_2)$，空间复杂度为 $O(mk_1+mk_2^2)$.

证明.　P_MFR 算法的正确性取决于式(4)～(14)的正确性.式(4)～(8)、(11)和(12)的证明在公式的引入时已经给出,下面证明其它公式.

在式(9)中,显然只有当 $right(r)\leq right(i)$ 时,$R(r,i,i)$ 才会有意义.令

I_1 表示 $R(*,r,i-1)$,

I_2 表示 $\max\limits_{d;d\in C(i),(d,r)\in S_{dk}(i)} R(d,r,i-1)$,

I_3 表示 $\max\limits_{k;k\in C(i),(r,k)\in S_{dk}(i),right(k)\leq right(i)} R(r,k,i-1)$.

首先证明 $\{i\}\cup I_1$,$\{i\}\cup I_2$,$\{i\}\cup I_2$ 都是以 r,i 为代表的 $\boldsymbol{M}(:,i)$ 的兼容行集.

根据定义,I_1 即 $R(*,r,i-1)$ 存在着一个划分,使得划分在同一个子集中的行彼此兼容,且其中一个子集的代表是行 r,另一个子集的代表为行 d 且 $right(d)\leq left(i)$.显然行 i 与行 d 没有共同覆盖的列,所以行 i 与 d 兼容,进而根据定理 5,行 i 与 d 所代表的子集中的所有行均兼容.

这样行 i 加入到 d 所在的子集后,行 i 将取代 d 作为该子集的代表,这样 $\{i\}\cup I_1$ 就是一个以 r,i 为代表的 $\boldsymbol{M}(:,i)$ 的兼容行集.

对于任意 $(d,r)\in S_{dk}(i)$,同样存在着 $R(d,r,i-1)$ 的一个划分,使得划分在同一个子集中的行彼此兼容,且其中一个子集的代表行是 r,另一个子集的代表为行 d.如果 $d\in C(i)$,即行 i 与 d 兼容,那么根据定理 5,行 i 与 d 所代表的子集中的所有行均兼容.行 i 加入到 d 所在的子集后,由于 $right(d)\leq right(r)\leq right(i)$,行 i 将取代 d 作为该子集的代表,显然 $\{i\}\cup R(d,r,i-1)$ 是一个以 r,i 为代表的 $\boldsymbol{M}(:,i)$ 的兼容行集,因此 $\{i\}\cup I_2$ 也是.

同理,对于任意 $(r,k)\in S_{dk}(i)$,$R(r,k,i-1)$ 存在着一个划分,使得划分到同一个子集中的行彼此兼容,且其中一个子集的代表是行 r,另一个为 k.如果 $k\in C(i)$,即行 i 与 k 兼容,根据定理 5,行 i 与 k 代表的子集中的所有行均兼容.行 i 加入到 k 所在的子集后,如果 $right(k)\leq right(i)$,行 i 将取代 k 作为该子集的代表行,这样 $\{i\}\cup R(r,k,i-1)$ 就是一个以 r,i 为代表的 $\boldsymbol{M}(:,i)$ 的兼容行集,因此 $\{i\}\cup I_3$ 也是.

现在证明 $\{i\}\cup\max(I_1,I_2,I_2)$ 是一个以 r,i 为代表的 $\boldsymbol{M}(:,i)$ 的最大兼容行集.

采用反证法,假设 $\{i\}\cup\max(I_1,I_2,I_3)$ 不是 $\boldsymbol{M}(:,i)$ 的一个以 r,i 为代表的最大兼容行集,这就是说 $R(r,i,i)$ 中的行数比 $\{i\}\cup I_1$、$\{i\}\cup I_2$ 和 $\{i\}\cup I_3$ 中的行数都要多.根据定义,$R(r,i,i)$ 中的行可以划分成两个子集,同一个子集的行互相兼容,且这两个子集中的行的代表分别是 r,i.令 i 所在的子集去掉行 i 后的代表行为 l,显然 l 与 i 兼容.

如果 $right(l)\leq left(i)$,则 $R(r,i,i)$ 去掉行 i 后得到的行集 $R(r,i,i)-\{i\}$ 是以 l,r 为代表的 $\boldsymbol{M}(:,i-1)$ 的一个兼容行集,如果假设成立,则 $R(r,i,i)-\{i\}$ 比 I_1 大,这与 $R(*,r,i-1)$ 的定义矛盾.

如果 $right(l)\geq left(i)$,那么或者 $(l,r)\in S_{dk}(i)$ 或者 $(r,l)\in S_{dk}(i)$(因为 $r\in S_k(i)$).如果 $(l,r)\in S_{dk}(i)$,则 $R(r,i,i)-\{i\}$ 是以 l,r 为代表的 $\boldsymbol{M}(:,i-1)$ 的一个兼容行集.如果假设成立,则 $R(r,i,i)-\{i\}$ 比 I_2 大,这与 $R(l,r,i-1)$ 的定义矛盾;如果 $(r,l)\in S_{dk}(i)$,则 $R(r,i,i)-\{i\}$ 是以 r,l 为代表的 $\boldsymbol{M}(:,i-1)$ 的一个兼容行集.如果假设成立,则 $R(r,i,i)-\{i\}$ 比 I_3 大,这与 $R(r,l,i-1)$ 的定义矛盾.

由此式(9)对于所有满足 $right(r)\leq right(i)$ 的 (r,i) 都成立,用同样的方法可以证明式(10)成立.

式(13)的证明:

令 I_1 表示 $\max\limits_{d;right(d)<left(i)} R(d,i,i)$,

I_2 表示 $\max\limits_{d;d\in S_k(i),right(d)\leq right(i),right(d)<left(i+1)} R(d,i,i)$.

根据定义,$R(*,,i,i)=\max(I_1,I_2)$,这样证明式(13)只需证明 $I_1=\max(\{i\}\cup R(*,*,i-1),\{i\}\cup\max\limits_{k;k\in C(i),right(k)\leq right(i)} R(*,k,i-1))=\{i\}\cup\max(R(*,*,i-1),\max\limits_{k;k\in C(i),right(k)\leq right(i)} R(*,k,i-1))$.

对于满足条件"$right(d)\leq left(i)$"的任意 d,用证明式(9)的方法可以证明下式成立:$R(d,i,i)=\{i\}\cup\max(R(*,d,i-1),\max\limits_{k;k\in C(i),right(k)\leq right(i)} R(d,k,i-1))$,因此

$$\begin{aligned}I_1&=\{i\}\cup\max\Big(\max_{d;right(d)<left(i)} R(*,d,i-1),\\&\qquad\max_{k;k\in C(i),right(k)\leq right(i)}\Big(\max_{d;right(d)<left(i)} R(d,k,i-1)\Big)\Big)\\&=\{i\}\cup\max\Big(R(*,*,i-1),\\&\qquad\max_{k;k\in C(i),right(k)\leq right(i)} R(*,k,i-1)\Big).\end{aligned}$$

式(13)成立.

式(14)的证明:

令 I_1 代表 $\max\limits_{k;right(k)<left(i)} R(*,k,i)$,

I_2 代表 $\max\limits_{k;k\in S_k(i)\cup\{i\},right(k)<left(i+1)} R(*,k,i)$.

根据定义,$R(*,*,i)=\max(I_1,I_2)$.对于满足 $right(k)\leq left(i)$ 的任意 $R(d,k,i)$ 而言,有 $right(d)\leq right(k)\leq left(i)$,因此有 $R(d,k,i)=$

$R(d,k,i-1)$（理由同式(8)），所以

$$I_1=\max_{k;right(k)<left(i)}\left(\max_{d;right(d)<left(i)}R(d,k,i)\right)$$
$$=\max_{k;right(k)<left(i)}\left(\max_{d;right(d)<left(i)}R(d,k,i-1)\right)$$
$$=\max_{k;right(k)<left(i)}R(*,k,i-1)=R(*,*,i-1).$$

式(14)成立.

算法的时间复杂度：算法的主体部分是步 3. 由于 $\boldsymbol{M}$ 经过了预处理，那么 $k\leq i$ 意味着 $left(k)\leq left(i)$，这时如果还有 $right(k)\geq left(i)$，则可以肯定行 k 覆盖列 $left(i)$，所以 $S_k(i)$ 中的元素个数 $|S_k(i)|$ 不会超过覆盖列 $left(i)$ 的行数. 由于 $\boldsymbol{M}$ 满足 (k_1,k_2) 参数化条件，所以 $|S_k(i)|\leq k_2$. 这样步 3.1 循环不超过 $(m-1)k_2$ 次，在步 3.1.3 中，判断两行是否兼容只需检查两行在它们共同覆盖的列上是否冲突，步 3.1.3 执行一次所需时间为 $O(k_1)$，所以步 3.1 所需时间为 $O(mk_1k_2)$；可以看出 $S_{dk}(i)$ 中的元组个数不会超过 k_2^2，所以步 3.3 和步 3.5 循环不超过 $(m-1)k_2^2$ 次，所需时间为 $O(mk_2^2)$；剩下的步 3.4、步 3.6 和步 3.7 所需时间为 $O(mk_1)$. 因此加上预处理，整个算法的时间复杂度为 $O(mk_2^2+mk_1k_2+m\log m+nk_2)$.

算法的空间复杂度：SNP 矩阵所需的空间为 $O(mk_1)$，记录所有的 $R(d,k,i)$ 需要的空间 $O(mk_2^2)$（只需记录相邻的两组 R 值，即 $i-1$ 和 i），所以算法的空间复杂度为 $O(mk_1+mk_2^2)$. 定理得证.

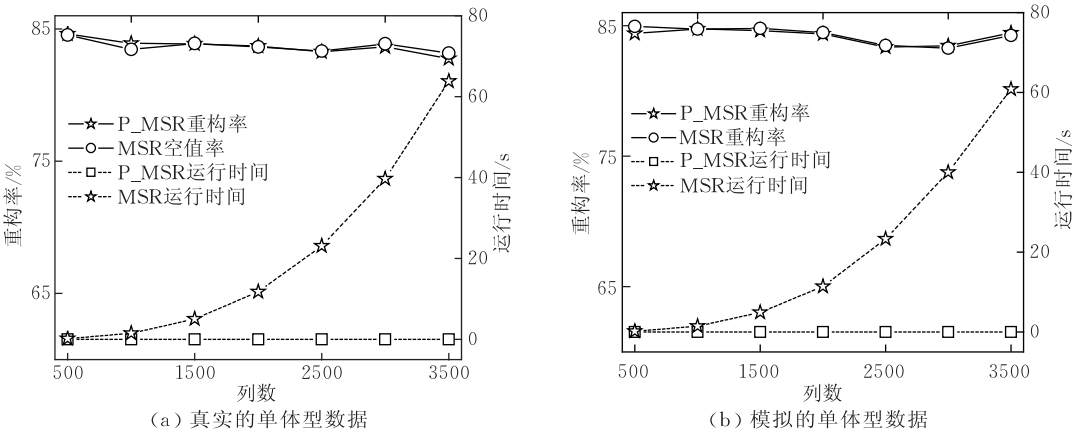
证毕.

4 实验结果

我们用 C++语言实现了 MFR（从文献[14]作者的源程序 Fast hare 中移植过来）、MSR、P_MFR 和 P_MSR 算法（源程序可通过 E-mail 向本文作者索取），在一台 Linux 服务器（4 个 Intel Xeon 3.6GHz CPU，4GB RAM）上对 MSR 和 P_MSR、MFR 和 P_MFR 算法的运行时间（Running time）和单体型重建率（Reconstruction rate）进行了比较. 单体型重建率指的是算法重建出的单体型中正确的 SNP 位点数与总的 SNP 位点数的比值[15].

实验中的单体型采用 2 种方式得到，第 1 种与文献[15]相同，采用来自公开数据库的真实的单体型，本文实验采用的真实单体型数据来自于国际人类基因组单体型图计划[16] 2006 年 7 月发布的数据文件 genotypes_chr1_CEU_r21_nr_fwd_phased.gz①，该文件中包含了 CEPH 样本（祖籍是北欧或西欧的美国犹他州人）中 60 个个体的单体型，每个单体型有 SNP 位点 193333 个，本文实验随机选择一个个体指定长度的一对单体型. 第 2 种与文献[8,14,17]一样用计算机模拟生成，即首先随机生成指定长度的单体型，根据指定的两个单体型的差异率来随机生成另一个单体型，本文采用差异率与文献[14]一样，为 20%.

由于原始的 DNA 片段测序数据很难得到，在得到一对单体型的基础上，上述文献均根据指定的参数利用计算机来随机生成片段数据集. 实验室中，Sanger 双脱氧链终止法的 DNA 测序误差约为 1%[18]，片段的覆盖度约为 10[13]. 为了使模拟生成的片段数据能很好地反映真实情况，与文献[14]一样，本文采用著名的 shotgun 测序模拟数据生成器 Celsim[19]. 下面的实验采用的测序误差为 1%，单体型长度，即 SNP 位点数，在一个区间内变化，生成片段的最小长度为 3，片段的最大长度和覆盖度在没有特别说明的情况下分别为7和10，生成的片段数



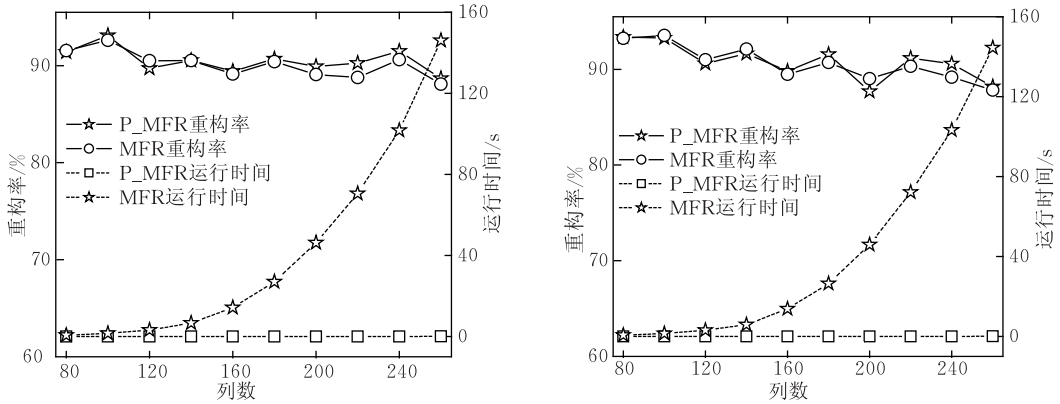
(a) 真实的单体型数据　　(b) 模拟的单体型数据

图 6 SNP 矩阵的列数变化时 P_MSR 和 MSR 的性能对比

① 数据来源：http://www.hapmap.org/downloads/phasing/2006-07_phaseII/

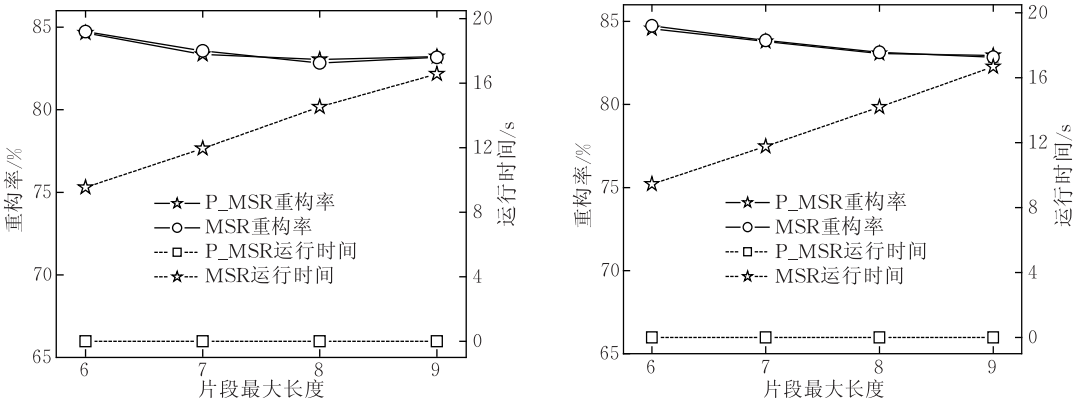
则按(单体型长度×片段覆盖度/片段平均长度)设置.模拟数据生成器的详细情况请参照文献[14,19].

图 6～图 11 的每一个点均为 100 次重复测试的平均值.下面各图中,图(a)均为在真实单体型数据上的实验结果,图(b)均为在模拟的单体型数据上的实验结果.因为在模拟的单体型数据上的实验结果与在真实单体型数据上的实验结果基本相同,所以下面主要讨论在真实单体型数据上的实验结果.每个子图中,左边的 Y 轴表示算法的重构精度,右边的 Y 轴表示算法的运行时间.



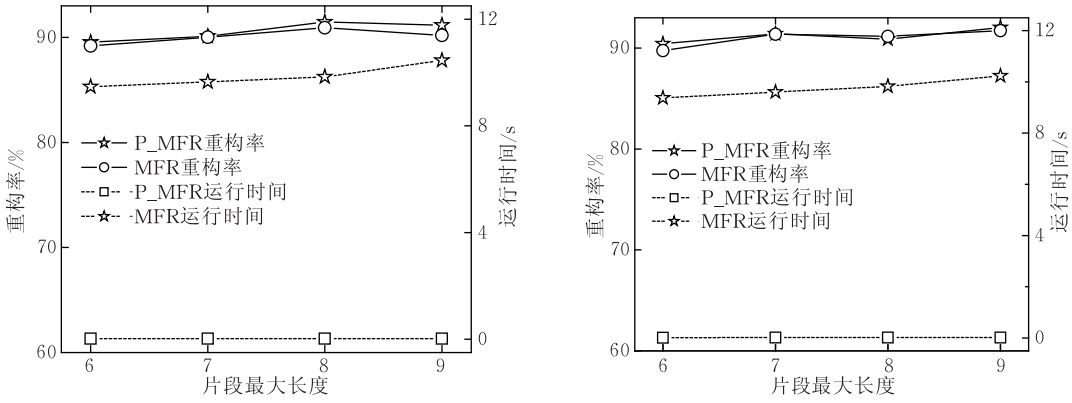
(a) 真实的单体型数据　　(b) 模拟的单体型数据

图 7　SNP 矩阵的列数变化时 P_MFR 和 MFR 的性能对比



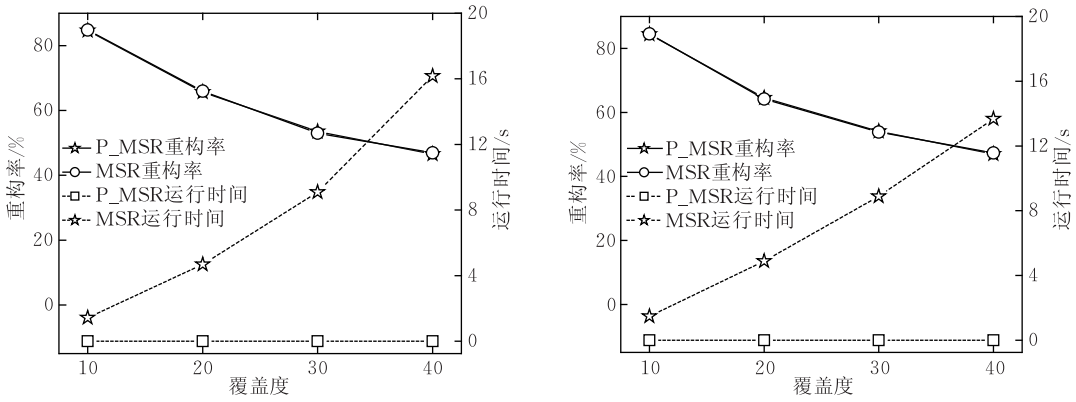
(a) 真实的单体型数据　　(b) 模拟的单体型数据

图 8　片段最大长度变化时 P_MSR 和 MSR 算法性能对比



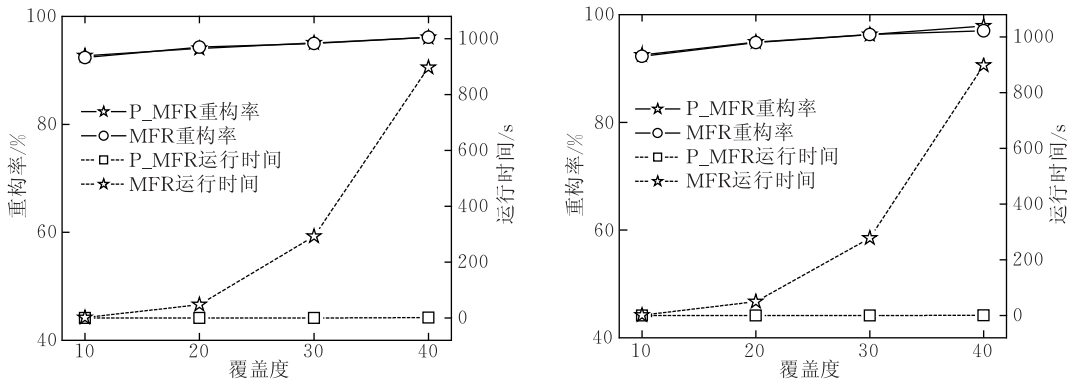
(a) 真实的单体型数据　　(b) 模拟的单体型数据

图 9　片段最大长度变化时 P_MFR 和 MFR 算法性能对比



(a) 真实的单体型数据 (b) 模拟的单体型数据

图 10 片段覆盖度变化时 P_MSR 和 MSR 的性能对比



(a) 真实的单体型数据 (b) 模拟的单体型数据

图 11 片段覆盖度变化时 P_MFR 和 MFR 的性能对比

图 6 和图 7 显示算法性能随 SNP 位点数(对应 SNP 矩阵的列数)变化的情况. 图 6(a)中,当位点数为 500 时,P_MSR 和 MSR 算法的单体型重构率分别是 84.62% 和 84.54%,运行时间为 0.0001s 和 0.191s;当位点数增加到 3500 时,P_MSR 和 MSR 算法的单体型重构率分别是 82.76% 和 83.18%,运行时间分别是 0.004s 和 63.82s.

图 7(a)中,当位点数为 80 时,P_MFR 和 MFR 算法的单体型重构率分别是 91.40% 和 91.55%,运行时间分别是 0.009s 和 0.67s;当位点数增加到 260 时,P_MFR 和 MFR 算法的单体型重构率分别是 88.68% 和 88.10%,运行时间分别是 0.055s 和 146.05s.

从图 6 和图 7 可以看出,随着 SNP 位点数的增加,算法的单体型重构精度有下降趋势;MFR 和 MSR 运行时间显著增长,这是因为当 SNP 位点数增加,覆盖度不变时,片断数也随着增加,所以 MFR 和 MSR 运行时间的增长速度是位点数增长速度的 3 次方,而 P_MFR 和 P_MSR 的时间则基本成线性增长.

在片段数和其它参数保持不变的条件下,图 8 和图 9 通过改变片段的最大长度和 SNP 位点数来比较各算法的性能. 图 8 的片段数保持 4000 不变. 图 8(a)中,在片段最大长度为 6、SNP 位点数为 1800 时,P_MSR 和 MSR 算法的单体型重构率分别是 84.6% 和 84.7%,运行时间分别是 0.001s 和 9.5s;当片段最大长度为 9、SNP 位点数为 2400 时,P_MSR 和 MSR 算法的单体型重构率分别是 83.2% 和 83.1%,运行时间分别是 0.001s 和 16.6s.

图 9 的片段数保持 300 不变. 图 9(a)中,在片段最大长度为 6、SNP 位点数为 135 时,P_MFR 和 MFR 算法的单体型重构率分别是 89.53% 和 89.19%,运行时间分别是 0.019s 和 9.47s;当片段最大长度为 9、SNP 位点数为 180 时,P_MFR 和 MFR 算法的单体型重构率分别是 91.17% 和 90.19%,运行时间分别是 0.025s 和 10.46s.

图 8 和图 9 的实验结果说明,当片段的最大长度和 SNP 位点数增加而片断数不变时,就单体型重构精度而言,P_MSR 和 MSR 算法有下降趋势,

P_MFR和 MFR 算法有上升趋势;就运行时间而言,MFR、P_MFR 及 P_MSR 的时间变化不大,而 MSR 有较大的变化,这是因为从时间复杂度理论分析上看,MSR 的运行时间应与位点数 n 的 2 次方成正比,而其他算法应与位点数 n 或片段的最大长度 k_1 成正比.

图 10 和图 11 则是在 SNP 位点数和其它参数保持不变的条件下,通过改变片段的覆盖度来改变片段数,从而比较各算法的性能. 在图 10(a)中,SNP 位点数保持 1000 不变. 当片断覆盖度为 10,片段数为 2000 时,P_MSR 和 MSR 算法的单体型重构率分别是 84.67% 和 84.86%,运行时间为 0.001s 和 1.44s;当片断覆盖度增加到 40,片段数为 8000 时,P_MSR 和 MSR 算法的单体型重构率分别是 46.55%和 46.96%,运行时间分别是 0.004s 和 16.17s.

在图 11(a)中,SNP 位点数保持 100 不变. 当片断覆盖度为 10、片段数为 200 时,P_MFR 和 MFR 算法的单体型重构率分别是 92.72%和 92.38%,运行时间分别为 0.01s 和 1.50s;当片断覆盖度增加到 40、片段数为 800 时,P_MFR 和 MFR 算法的单体型重构率分别是 96.10%和 96.13%,运行时间分别是 0.99s 和 897.14s.

图 10 和图 11 的实验结果说明,当位点数保持不变的条件下,片段覆盖度增加,片段数增加,SNP 位点的最大覆盖度(k_2)也会随之增加,就单体型重构精度而言,P_MSR 和 MSR 算法性能显著下降,P_MFR和 MFR 算法性能上升;就运行时间而言,P_MSR及 P_MFR 的时间变化不大,MSR 基本上呈线性变化,但 MFR 的运行时间显著上升. 这次实验结果也说明了算法实际运行时间与其时间复杂度理论分析是吻合的(MFR 的运行时间与片段数 m 的 3 次方成正比,MSR 和 P_MSR 只与片段数 m 或最大覆盖度 k_2 成正比,而 P_MFR 与最大覆盖度 k_2 的平方成正比,但由于 k_2 的值比片段数小得多,所以P_MFR所需的时间仍然很小).

上述实验结果表明,无论是对于真实的单体型数据还是模拟的单体型数据,P_MFR(或 P_MSR)在运算速度上明显优于 MFR(或 MSR);在单体型重构率上,P_MFR 和 MFR(或 P_MSR 和 MSR)没有实质的差别,这是因为 P_MFR 和 MFR(或P_MSR 和 MSR)都是精确算法,即对于保留的行数(或列数)而言,它们的结果是完全相等的,尽管在保留的具体行(或列)上可能存在随机性,算法的单体型重构精度不完全耦合,但是总体上来说没有优劣之分.

5　结论和展望

同一物种不同个体的遗传差别来自于单核苷酸多态性,对单体型的识别对遗传病相关基因的定位、药物的设计有很重要的作用,是身份识别和亲子鉴定的有效手段. 本文根据基因组测序中能直接测序 DNA 片断长度的最大长度、覆盖同一 SNP 位点的最大片断数与测序区域的总长度、测序得到的总片断数相比,他们均很小的特点,对个体单体型问题进行了参数化. 当 DNA 片断数为 m,每个片段覆盖的 SNP 位点数不超过 k_1,测序区域的 SNP 位点数为 n,覆盖同一 SNP 位点的片段数不超过 k_2 的情况下,本文提出的算法使求解无空隙的 MSR 和 MFR 问题的时间复杂度分别降低到 $O(nk_1k_2+m\log m+mk_1)$ 和 $O(mk_2^2+mk_1k_2+m\log m+nk_2)$,和文献[5]中的时间复杂度为 $O(mn^2)$ 和 $O(m^2n+m^3)$ 的算法相比,效率提高了很多,具有很好的可扩展性和较高的实用价值. 对于有空隙的 SNP 矩阵的个体单体型 MSR 和 MFR 问题及个体单体型问题的其他计算模型,绝大部分被证明是 NP 难的,求解这些问题的现有算法都是指数时间复杂度,对于这些问题,本文的参数化方法提供了一个很好的方向.

致　谢　感谢 Panconesi A、Sozio M 及 Myers M,他们慷慨地提供了 Fast hare 和 Celsim 的源代码,也感谢匿名审稿人对本文初稿的宝贵意见!

参　考　文　献

[1] Kruglyak L, Nickerson D A. Variation is the spice of life. Nature Genetics, 2001, 27(3): 234-236

[2] Stephens J C, Schneider J A, Tanguay D A et al. Haplotype variation and linkage disequilibrium in 313 human genes. Science, 2001, 293(5529): 489-493

[3] Horikawa Y, Oda N, Cox N J et al. Genetic variation in the gene encoding calpain-10 is associated with type 2 diabetes mellitus. Nature Genetics, 2000, 26(2): 163-175

[4] Lancia G, Bafna V, Istrail S, Lippert R, Schwartz R. SNPs problems, complexity and algorithms//Proceedings of the 9th Annual European Symposium on Algorithms (ESA 2001). Aarhus, Denmark, 2001: 182- 193

[5] Bafna V, Istrail S, Lancia G, Rizzi R. Polynomial and APX-hard cases of the individual haplotyping problem. Theoretical Computer Science, 2005, 335(1): 109-125

[6] The International SNP Map Working Group. A map of human genome sequence variation containing 1.42 million single

nucleotide polymorphisms. Nature, 2001, 409(6822): 928-933

[7] International Human Genome Sequencing Consortium. Initial sequencing and analysis of the human genome. Nature, 2001, 409(6822): 860-921

[8] Wernicke S. On the algorithmic tractability of single nucleotide polymorphism (SNP) analysis and related problems [Ph. D. dissertation]. Tübingen: Universität Tübingen, 2003

[9] Gray I C, Campbell D A, Spurr N K. Single nucleotide polymorphisms as tools in human genetics. Human Molecular Genetics, 2000, 9(16): 2403-2408

[10] Sanger F, Nicklen S, Coulson A R. DNA sequencing with chain-terminating inhibitors. Proceedings of the National Academy of Sciences of the USA, 1977, 74(12): 5463-5467

[11] Hinds D A, Stuve L L, Nilsen G B et al. Whole-genome patterns of common DNA variation in three human populations. Science, 2005, 307(5712): 1072-1079

[12] Gabriel S B, Schaffner S F, Nguyen H et al. The structure of haplotype blocks in the human genome. Science, 2002, 296(5576): 2225-2229

[13] Weber J L, Myers E W. Human whole-genome shotgun sequencing. Genome Research, 1997, 7(5): 401-409

[14] Panconesi A, Sozio M. Fast hare: A fast heuristic for single individual SNP haplotype reconstruction//Proceedings of the 4th International Workshop on Algorithms in Bioinformatics (WABI 2004). Bergen, Norway, 2004: 266-277

[15] Wang Rui-Sheng, Wu Ling-Yun, Li Zhen-Ping, Zhang Xiang-Sun. Haplotype reconstruction from SNP fragments by minimum error correction. Bioinformatics. 2005, 21(10): 2456-2462

[16] The International HapMap Consortium. The international HapMap project. Nature, 2003, 426(6968): 789-796

[17] Hüffner F. Algorithm engineering for optimal graph bipartization//Proceedings of the 4th International Workshop on Experimental and Efficient Algorithms (WEA 2005). Santorini Island, Greece, 2005: 240-252

[18] Ansorge W, Voss H, Wirkner U et al. Automated sanger DNA sequencing with one label in less than four lanes on gel. Journal of Biochemical and Biophysical Methods, 1989, 20(1): 47-52

[19] Myers G. A dataset generator for whole genome shotgun sequencing//Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology (ISMB 1999). Heidelberg, Germany, 1999: 202-210

XIE Min-Zhu, born in 1969, Ph. D., associate professor. His major research interests include bioinformatics, exact and parameterized algorithms.

CHEN Jian-Er, born in 1954, Ph. D., professor, Ph. D. supervisor. His major research interests include bioinformatics, exact and parameterized algorithms, computer graphics, and computer networks.

WANG Jian-Xin, born in 1969, Ph. D., professor, Ph. D. supervisor. His major research interests include computer networks, computational optimization algorithms and bioinformatics.

Background

This work is supported by the National Natural Science Foundation of China (No. 60773111), the National Basic Research Program (973 Program) of China (No. 2008CB317107), Program for Changjiang Scholars and Innovative Research Team in University (No. IRT0661), Hunan Provincial Natural Science Foundation of China (No. 09JJ3116), China Postdoctoral Science Foundation and Central South University Postdoctoral Science Foundation.

Haplotypes are widely used in complex diseases association study. However, it is time-consuming and expensive to determine haplotypes only using biological techniques. Therefore, effective computational techniques are in urgent demand to help determine haplotypes. Recently more and more individual genomes have been sequenced and the individual haplotyping problem, i. e., using computational techniques to infer the haplotypes of an individual from his or her DNA sequence fragments, has been a hotspot of bioinformatics.

Lancia et al. introduced two optimal computational models of the individual haplotyping problem: Minimum SNP removal (MSR) and Minimum fragment removal (MFR), and they proved that MFR and MSR are NP-hard if fragments have gaps. When all fragments are gapless, Bafna et al. introduced an $O(mn^2)$ algorithm and an $O(m^2n+m^3)$ algorithm to solve MSR and MFR respectively, where m is the number of fragments and n is the number of SNP sites. When inferring a pair of haplotypes over a large region of a chromosome, m and n are very large, and both algorithms are still slow.

Limited by current sequencing techniques, the maximum length of fragments directly sequenced by modern automated DNA sequencing instruments is small. In order to save money and time, in the real DNA sequence fragments data, the number of fragments covering a SNP site is limited. Based on the above observations, the current paper parameterized the models of MSR and MFR and introduced new parameterized algorithms P_MSR and P_MFR, which solve the gapless MSR and MFR problems with significantly improved efficiency respectively.