ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.18, No.9, September 2007, pp.2070−2082
DOI: 10.1360/jos182070

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

有Mate-Pairs的个体单体型MSR问题的参数化算法*

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,湖南 长沙 410083)

[2](湖南师范大学 物理与信息科学学院,湖南 长沙 410081)

Parameterized Algorithm of the Individual Haplotyping MSR Problem with Mate-Pairs

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, China)

[2](College of Physics and Information Science, Hu'nan Normal University, Changsha 410081, China)

+ Corresponding author: Phn: +86-731-8830212, E-mail: jxwang@mail.csu.edu.cn

Xie MZ, Chen JE, Wang JX. Parameterized algorithm of the individual haplotyping MSR problem with Mate-Pairs. *Journal of Software*, 2007,18(9):2070−2082. http://www.jos.org.cn/1000-9825/18/2070.htm

Abstract: The individual haplotyping MSR (minimum SNP removal) problem is the computational problem of inducing an individual's haplotypes from one's DNA fragments sequencing data by dropping minimum SNPs (single-nucleotide polymorphisms). To solve the problem, Bafna, *et al*. had provided an algorithm of time complexity $O(2^k n^2 m)$ with the number of fragments m, the SNP sites n, the maximum number of holes k in a fragment. In the case that there are some Mate-Pairs, since the number of holes in a Mate-Pair can reach 100, Bafna's algorithm is impracticable. Based on the characters of DNA fragments, this paper presents a new algorithm of time complexity $O((n-1)(k_1-1)k_2 2^{2h}+(k_1+1)^{2h}+nk_2+mk_1)$ with the maximum number of SNP sites that a fragment covers k_1 (no more than n), the maximum number of the fragments covering a SNP site k_2 (usually no more than 19) and the maximum number of fragments covering a SNP site whose value is unknown at the SNP site h (no more than k_2). Since the time complexity is not directly related with k, the algorithm can deal with the MSR problem with Mate-Pairs efficiently, and is more scalable and applicable in practice.

Key words: SNPs; genotype; haplotype; parameterized algorithm; computational complexity

摘 要: 个体单体型MSR(minimum SNP removal)问题是指如何利用个体的基因测序片断数据去掉最少的SNP(single-nucleotide polymorphisms)位点,以确定该个体单体型的计算问题.对此问题,Bafna等人提出了时间复杂度为$O(2^k n^2 m)$的算法,其中,m为DNA片断总数,n为SNP位点总数,k为片断中洞(片断中的空值位点)的个数.由于一个Mate-Pair片段中洞的个数可以达到 100,因此,在片段数据中有Mate-Pair的情况下,Bafna的算法通常是不可行的.根据片段数据的特点提出了一个时间复杂度为$O((n-1)(k_1-1)k_2 2^{2h}+(k_1+1)^{2h}+nk_2+mk_1)$的新算法,其中,$k_1$为一个片断覆

* Supported by the National Natural Science Foundation of China under Grant No.60433020 (国家自然科学基金); the Program for New Century Excellent Talents in University of China under Grant No.NCET-05-0683 (新世纪优秀人才支持计划); the Program for Changjiang Scholars and Innovative Research Team in University of China under Grant No.IRT0661(国家教育部创新团队资助项目); the Scientific Research Fund of Hunan Provincial Education Department of China under Grant No.06C52 (湖南省教育厅资助科研项目)

Received 2006-09-23; Accepted 2006-12-19

盖的最大SNP位点数(不大于n),k_2为覆盖同一SNP位点的片段的最大数(通常不大于 19),h为覆盖同一SNP位点且在该位点取空值的片断的最大数(不大于k_2).该算法的时间复杂度与片断中洞的个数的最大值k没有直接的关系,在有Mate-Pair片断数据的情况下仍然能够有效地进行计算,具有良好的可扩展性和较高的实用价值.

关键词: 单核苷酸多态性;基因型;单体型;参数化算法;计算复杂度

中图法分类号: TP301 **文献标识码:** A

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,对疾病有不同的抵抗能力,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大约有 340 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,也可用于人类各群体的遗传关系分析.Stephens等人采用个体单体型问题变异的方法研究人类 313 个基因中的 3 899 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等人[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与 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.对于人类等二倍体生物,染色体是成对存在的,每一对染色体的 DNA 序列除 SNP 位点外都是一样的.在一条染色体 SNP 位点上的碱基序列叫做单体型(haplotype).对于任何一个二倍体生物,都有一对单体型.单体型在 SNPs 的上述应用中扮演着重要的角色.但是,在当前的实验技术下,直接测定个体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基因测序片断数据的特点,尤其是在片段数据中包含有 Mate-Pair 片段数据时,对确定该个体的单体型的一个计算优化模型 MSR 进行参数化求解.本文第 1 节介绍个体单体型问题的抽象模型和当前的研究现状.第 2 节阐述 MSR 的参数化算法,分析其复杂度.第 3 节对实验结果进行分析.最后进行总结.

1 个体单体型问题及当前研究现状

像人类等双倍体生物,DNA 序列是按染色体成对出现的.对于任意一个 SNP 位点来说,一对染色体上的碱基可以是相同的,这种现象叫做纯合(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 2 个字母的字符集$\{A,B\}$上的字符序列来表示,不必用真正的碱基字符.一对染色体在 SNP 位点的投影序列则是一对这样的序列,叫做基因型(genotype).

随着对 SNPs 的深入研究,Lancia 等人在文献[4]中提出直接利用个体的 DNA 片断数据来确定个体的一对单体型,这个计算问题就是本文所说的个体单体型问题(individual haplotyping problem).

在对基因组进行测序时,限于技术上的局限,实验室只能直接对较短的DNA片断进行测序,这些片断来自一对染色体的不同单体,测序过程中也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少数据[4],以便能够发现一对单体型与剩下的数据兼容,其中,去掉的数据可以是片断或SNP位点.

n个SNP位点按在染色体上的次序从左到右记作S:$\{1,2,...,n\}$,m个片断记作F:$\{1,2,...,m\}$,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'或'–',其中,'–'表示片断在该位点的取值为空(其值未能确定),或者说,该片断在这个SNP位点上有一个洞(hole).DNA片断的数据集就可以表示为在$\{A,B,–\}$上的一个$m\times n$的矩阵,叫做SNP矩阵$\boldsymbol{M}$[4],其中,矩阵$\boldsymbol{M}$的行表示片断,列表示SNP位点,$\boldsymbol{M}_{ij}$的值表示第i个

片断在第j个SNP位点上的取值,而取$\boldsymbol{M}$的前j列所构成的矩阵记作$\boldsymbol{M}[:,j]$.

对于一个 SNP 矩阵 $\boldsymbol{M}$ 有如下概念或定义:

对于 $\boldsymbol{M}$ 的两行,如果它们在列 j 上的值不相等,且都不为'−',则这两行在列 j 上发生冲突;如果这两行在所有的列上均不冲突,则这两行互相兼容.

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

也就是说,当$\boldsymbol{M}_{i,j}$的值不为'−',或者列j前后至少各有一列,使得行i在这两列上的值均不为'−'时,行i覆盖列j,列j被行i覆盖.从定义可以看出,任意一行所覆盖的列是连续的,这些列中序号最小的列称为该行的起始列,序号最大的列称为该行的终止列.

这样,从起始列到终止列,该行上取'−'值的列数就是该行对应的片断中洞的个数.

定义 2($\boldsymbol{M}$ 是可行的). $\boldsymbol{M}$ 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.

显然,一个 SNP 矩阵 $\boldsymbol{M}$ 是可行的当且仅当可以找到一对单体型,使得 $\boldsymbol{M}$ 中的任意一行总是可以与其中的一个单体型兼容(其中单体型表示为$\{A,B\}$上的长为 n 的字符序列).

Lancia 等人在文献[4]中最初引入了 MSR 问题:

最少 SNP 删除(minimum SNP removal,简写 MSR)问题.给定一个 SNP 矩阵 M,删除最少的列(SNP 位点)使 M 可行.

为了提高DNA测序片段组装的精确度,在双管(double-barreled)shotgun测序中[5],对一个较长的片段两端进行测序,中间留下一片空隙(gap,一连串的洞).另外,在实际测序中,由于各种原因,一个DNA片段中的某些碱基无法确定,因而在片段中也可能出现空隙.

文献[6]证明:如果一片断中有空隙时,MSR是APX-hard,对于m个DNA片断,n个SNP位点,一个片段中至多有k个洞的MSR问题,该文提出了时间复杂度为$O(2^k n^2 m)$、空间复杂度为$O(2^k n^2)$的算法.

人类基因组上SNP的分布密度约为 0.1%[7−9],目前已知的SNP位点多达 300 多万个[1,7−10],而Celera公司对人类基因组的常染色体部分进行测序时得到的片断数目为 27 271 853 个[9].由此可以看出,为了推断出染色体的一个区域的单体型,需要处理的片断数m和SNP位点数n都很大.

当前,DNA测序的主导方法是Sanger双脱氧链终止法[11].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 800~1 200 个碱基.各大测序中心使用的第三代测序仪如ABI 3730,MageBACE等可直接测定的片断的最大长度约为 1 000 个碱基.采用了双管测序法可以从两端对一段较长的DNA克隆进行测序,得到位于克隆两端的长度在 1KB以内,彼此间距离已知的两个读段,这两个读段构成一个Mate-Pair.根据克隆的载体不同,Mate-Pair的典型长度有 2KB,10KB,50KB和 150KB[12].对DNA片断进行测序时,由于各种原因无法确定某些位点的碱基值或是其中有Mate-Pairs数据,DNA片断数据中就会出现空隙.国际人类基因组和Celera公司对人类基因组测序时都大量用到Mate-Pairs.如果片段测序数据中出现的Mate-Pair的长度达到了 150KB,在这种情况下,片段中洞的个数则会达到 100 个以上(平均 1KB有 1 个SNP位点[7−9]).显然,文献[6]的以片段中的洞的最大数k为参数的时间复杂度为$O(2^k n^2 m)$、空间复杂度为$O(2^k n^2)$的算法是不可行的.因而,根据目前基因组测序的技术现状,针对DNA片断的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.

2 有 Mate-Pairs 片断数据的 MSR 参数化算法

在有Mate-Pair片断数据的情况下,虽然Mate-Pair片断中洞的最大数可达 100 个,但是,因为测序的成本和所需的时间与覆盖率成正比,所以,实验室测序时片段的覆盖率(coverage)不是很大.如Celera公司的人类基因组测序工程中的片断覆盖率为 5.11[9]、国际人类基因组测序联盟(IHGSC)在人类基因组计划中的片断数据的覆盖率是 4.5[7](这些片断中有些是Mate-Pair片断数据).虽然在人类基因组的每个位置的片断覆盖程度并不完全相同,但是,文献[12]对文献[9]中的片断数据进行了详细的分析,从其片断覆盖图上可以看出,绝大部分位点上的片断数为 5 左右,最大值没有超过 19.由此可以看出,覆盖一个SNP位点的片断数远小于片断总数.覆盖某一SNP位点的所有片断中,在该位点上取空值的片断数肯定不大于覆盖该位点的片断总数.另外,一个片断覆盖的SNP位点

数一般比SNP位点总数要小(尽管SNP分布不均匀,从已有的数据来看,长为 150KB的Mate-Pair,其覆盖的SNP位点数约为 150 个左右[13,14]).

根据以上事实,特提出以下参数化条件:

定义 3((k_1,k_2,h)参数化条件). k_1,k_2,h是正整数,(k_1,k_2,h)参数化条件定义为任意片段覆盖的SNP位点数不超过k_1(从第 1 个非'–'值SNP位点到最后一个非'–'值SNP位点之间最多只有k_1–2 个位点,中间允许出现任意多个洞),对于任意一个SNP位点,覆盖该位点的片段数不超过k_2,而其中在该位点上取空值的片段数不会超过h.

对于一个SNP矩阵,满足(k_1,k_2,h)参数化条件就是该矩阵的任意一行覆盖的列数不超过k_1,即该行取非'–'值的两列之间最多有k_1–2 列,覆盖任意一列的行数不超过k_2,其中在该列上的值为'–'值的行数最多为h.

图 1 给出了一个满足(9,6,2)参数化条件的 11 行×10 列的 SNP 矩阵,其中,覆盖列数最多的为第 2 行,覆盖了从列 2~列 10 的 9 个列;覆盖第 6 列的行数最多,为 6;对于任意一列,覆盖该列的行中最多只有 2 行在该列上取'–'值.

对于任意一个m×n的SNP矩阵$\boldsymbol{M}$,对$\boldsymbol{M}$的所有行进行一次扫描,就可以统计出每行覆盖的列数、覆盖每列的行数及覆盖每列且在该列取'–'值的行数,从每行覆盖的列数中取最大值就得到k_1(k_1≤n),从覆盖每列的行数中取最大值就得到k_2(k_2≤m),同样,从覆盖每列且在该列取'–'值的行数中取最大值就得到h(h≤k_2),这样,$\boldsymbol{M}$就满足(k_1,k_2,h)参数化条件.在极端的情况下,k_1=n,k_2=m,h=k_2;但根据本节第 1 段对已有的基因组测序片断数据的分析,从文献[12]可知,对Celera公司的这些片断数据对应的SNP矩阵而言,k_2=19,而h不会大于k_2.

根据包含 Mate-Pair 的片断数据的特点,本文对 MSR 问题进行参数化:

PM_MSR(parameterized minimum SNP removal with Mate-Pair)问题.给定一个满足(k_1,k_2,h)参数化条件的SNP矩阵$\boldsymbol{M}$,PM_MSR问题是要求删除最少的列(SNP位点),也就是保留最多的列使$\boldsymbol{M}$可行.

本文下面所述的 PM_MSR 的解是指能保留下来的最大列数.

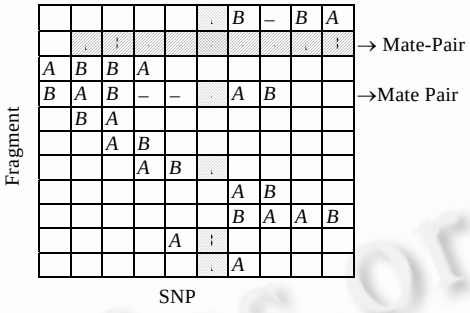

Fig. 1 An SNP matrix satisfying (9,6,2) parameterized condition

图 1 一个满足(9,6,2)参数化条件的 SNP 矩阵

2.1 预处理

对任意一个 SNP 矩阵 $\boldsymbol{M}$,我们首先进行如下预处理:

Step 1. 得出覆盖各列的行号的有序集.覆盖第 j 列的行号的有序集记作 *rowset*(j).

Step 2. 去掉冗余列.对 SNP 矩阵的每一列 j,如果 *rowset*(j)中的所有行在该列的取值中没有出现'*A*'或出现'*B*',则标记该列.最后去掉所有标记的列.

Step 3. 去掉冗余行.对 SNP 矩阵的每一行,如果该行取非空值的列在 Step 2 被全部删除,则标记该行.最后去掉所有标记的行,修改受影响的有序集 *rowset*.

预处理的时间复杂度分析:如果SNP矩阵采用压缩的形式存储,如每一个行记录其起始列号和终止列号,再记录该行在其覆盖的列上的值(可以有其他压缩的存储方式,但文中其余部分所指的压缩存储形式隐指这种),那么对所有行扫描一遍可以得到各列的有序集*rowset*,Step 1 所需时间为$O(mk_1)$.

判断列j是否冗余,可以对*rowset*(j)中的行在列j上的取值进行统计,所需的时间为$O(k_2)$,这样标记出所有的

冗余列需时间$O(nk_2)$.去掉冗余列,可以先从列 1 到列n扫描一遍,累计每一列前面有多少列被标记,所需时间为$O(n)$,然后对所有的行扫描一遍,去掉被标记的列上的值,对片断的起始列号和终止列号进行修改,所需时间为$O(mk_1)$.行的起始列号和终止列号的修改方法为:起始列号减去起始列前被标记的列数得到新的起始列号;如果终止列没有被标记,则终止列号减去终止列前被标记的列数得到新的终止列号.如果被标记,则再减 1,如果片断所覆盖的所有列都被标记,则该片断新的终止列号就会比新的起始列号小 1.这样,Step 2 所需时间为$O(nk_2+mk_1)$,然后再对每一列的*rowset*中的行扫描一次,对行的起始列号和终止列号进行比较,如果是冗余行,则从*rowset*中去掉,这样,Step 3 所需时间为$O(nk_2)$.所以,预处理所需的时间为$O(nk_2+mk_1)$.

如果不是采用压缩的形式存储,预处理的时间复杂度则为 $O(mn)$.

定理 1. 一个满足(k_1,k_2,h)参数化条件SNP矩阵$\boldsymbol{M}$经过以上的预处理后得到的SNP矩阵$\boldsymbol{M}'$满足(k_1,k_2,h')参数化条件,其中,$h'=\min(\max(k_2-2,0),h)$.

证明:去掉冗余列和冗余行的预处理显然不会增加对应片段的长度、覆盖某一SNP位点的片断数和覆盖某一位点并在该位点取空值的片断数.而经过预处理以后,任何一列对应的SNP位点上至少应有一个片断取‘*A*’值,还有一个取‘*B*’值,所以,覆盖任意SNP位点的取‘–’值的片断数不会大于k_2-2,当然也不会比 0 小. □

定理 2. 令一个 SNP 矩阵 $\boldsymbol{M}$ 经过以上的预处理得到的 SNP 矩阵为 $\boldsymbol{M}'$,其中,去掉的列的集合为 X,去掉的行的集合为 Y,则 $\boldsymbol{M}$ 的 PM_MSR 的解(这里是指能保留的最大列数)等于 $\boldsymbol{M}'$的 PM_MSR 的解与 X 的列数之和.

证明:用顶点表示行,边表示两行冲突,则一个 SNP 矩阵对应一个冲突图.由文献[4]可知,一个 SNP 矩阵是可行的当且仅当其对应的冲突图是一个二部图.在预处理中某一列被去掉,是因为不存在两行使得其中一行在该列上的值为‘*A*’,另一行为‘*B*’,即不存在两行在该列上发生冲突,所以任意两行在去掉的列上是不冲突的.而 Y 中的行在 X 外的列上的取值都为空,所以,Y 中的任意行均不与其他行发生冲突.如果 $\boldsymbol{M}'$的 PM_MSR 的解所对应的保留列上的冲突图是二部图,那么加上 Y 中的所有行和 X 中的所有列后,该冲突图不会增加新的边,仍然是二部图.所以,$\boldsymbol{M}$ 的 PM_MSR 的解不小于 $\boldsymbol{M}'$的 PM_MSR 的解与 X 中的列数之和;反之,如果 $\boldsymbol{M}$ 的 PM_MSR 的解所对应的保留列上的冲突图是二部图,那么,去掉那些代表 Y 中的行的顶点,再删除由于去掉 X 中的列后不再冲突的行之间的边后,冲突图必定还是二部图.所以,$\boldsymbol{M}'$的 PM_MSR 的解不小于 $\boldsymbol{M}$ 的 PM_MSR 的解与 X 的列数之差.因此,$\boldsymbol{M}$ 的 PM_MSR 的解等于 $\boldsymbol{M}'$的 PM_MSR 的解与 X 的列数之和.定理得证. □

从定理 2 可知,求解一个SNP矩阵的PM_MSR的解可以归结为求其预处理后形成的矩阵的对应解.为了叙述简便,我们称一个经过上述预处理后的SNP矩阵为精简的SNP矩阵.本节下面算法的输入均为预处理后得到的精简的SNP矩阵$\boldsymbol{M}$,该矩阵具有如下特点:任何一个列总有一些行的值是‘*A*’,还有一些行的值为‘*B*’.

2.2 PM_MSR算法

对于一个精简的SNP矩阵$\boldsymbol{M}$,因为在列j上,有些行取‘*A*’值也有些行取‘*B*’值,所以,如果列j被保留,要使$\boldsymbol{M}$可行,则覆盖列j的行肯定必须划分成两个子集,并且在列j上的取值为‘*A*’的行应该划分到同一个子集;而取值为‘*B*’的行肯定应该划分到另一个子集中.这样,只有在该列取‘–’值的行的划分才是不确定的.显然,对于一个满足(k_1,k_2,h)参数化条件的精简的SNP矩阵$\boldsymbol{M}$,覆盖列j且在列j上的值为‘–’的片段数最多为h,那么,覆盖列j的全体片段上所有可行的划分数最多是2^h.

定义 4. 划分函数:对某一列j,令覆盖列j的所有行的序号集为F,列j上的划分函数P定义为满足如下条件的映射$F\to\{A,B\}$:对于覆盖列j的行i,如果$\boldsymbol{M}_{i,j}$=‘*A*’,则$P(j)$=‘*A*’;如果$\boldsymbol{M}_{i,j}$=‘*B*’,则$P(j)$=‘*B*’;如果$\boldsymbol{M}_{i,j}$=‘–’,则$P(j)$可从‘*A*’和‘*B*’中任选一个.

对于一个满足(k_1,k_2,h)参数化条件的精简的SNP矩阵$\boldsymbol{M}$,任意列上的划分函数的个数最多是2^h.

从定义4可以看出,按照P函数值对覆盖列j的行进行划分,函数值相同的行划分给同一子集;反之,对覆盖列j的行的任意划分,如果划分出来的每个子集中的行互不冲突,则该划分必定对应着列j上的一个划分函数.

定义 5. $K[j,P]$和 $Column[j,P]$:P 为列 j 上的划分函数,$K[j,P]$定义为满足下列条件从列 1 到列 j 中能保留下的最多列数,对应的保留下来的列的集合定义为 $Column[j,P]$:列 j 必须保留;对 $\boldsymbol{M}[:,j]$中的所有行存在着一个划分,使得划分在同一子集中的行在 $Column[j,P]$中的所有列上均不冲突,且划分在同一子集中的行,如果其覆盖列 j,

则其 P 函数值必相同.

显然,保留 $Column[j,P]$中的列可使 $\boldsymbol{M}[:,j]$可行.对于一个精简的 SNP 矩阵第 1 列而言,覆盖第 1 列的行在第 1 列的值都是确定的,所以片段的划分函数 P 是唯一的,且有

$$K[1,P]=1;Column[1,P]=\{1\} \tag{1}$$

定义 6. $OK(j,P)$:P为列j上的划分函数,$OK(j,P)$定义为所有满足下列条件的(j',P')的集合:P'为列j'上的划分函数且$\max(0,j-k_1)<j'<j$;P'和P兼容,就是P'和P对既覆盖j'又覆盖j的行的划分不发生冲突,即对任意既覆盖j'又覆盖j的两行,如果它们的P'函数值相等,则P函数值必相等,这表示由P'划分在同一子集中的行,按P来划分也必须在同一子集中.

对于一个满足(k_1,k_2,h)参数化条件的精简的SNP矩阵$\boldsymbol{M}$,根据$K[j,P]$的定义,有下面的递推式(正确性见定理 3 的证明):

$$K[j,P]=\max(A[j-k_1],\ \max_{(j',P')\in OK(j,P)}(K[j',P']))+1 \tag{2}$$

其中,$A[j]$为满足 $j'\le j$ 的所有 $K[j',P']$的最大值,即

$$A[j]=\begin{cases}0, & \text{if } j<1\\ 1, & \text{if } j=1\\ \max\limits_{\text{all possible } K[j',P'] \text{ with } j'<j} K[j',P'], & \text{if } j>1\end{cases} \tag{3}$$

从式(1)~式(3)可以看出,对任意 j,$A[j]$不会比 0 小,且 $m\times n$ 矩阵 $\boldsymbol{M}$ 的 PM_MSR 解就是 $A[n]$.

根据式(1)~式(3)可以得到 PM_MSR 算法,见算法 1.

算法 1. PM_MSR.

Input: 一个精简的 SNP 矩阵 $\boldsymbol{M}$;

Output: $\boldsymbol{M}$ 的 MSR 解.

Step 1: 对$\boldsymbol{M}$中所有的片断扫描一次,得到$\boldsymbol{M}$的行列数m,n及其 3 个参数k_1,k_2,h

Setp 2: //初始化

Setp 2.1: for j=0,...,k_1 do

Setp 2.1.1: $A[j]$=0; $H[j]$=0;

Setp 2.1.2: for P=0,...,2^h–1 do $K[j,P]$=0;

Setp 2.2: $curColum$=0, j=0, K[0,0]=1; //根据式(1),$curColum$=0 表示第 1 列

Step 3: while $curColum$<n–1 do //循环 n–1 次

Step 3.1: $curColum$++; j=(j+1) mod (k_1+1)

Step 3.2: $H[j]$=覆盖列 $curColum$ 的行中在 $curColum$ 上取'-'值的行数

Step 3.3: for P=0,...,$2^{H[j]}$–1 do //P是一个在$curColum$列上的划分函数

Step 3.3.1: $K[j,P]=A[(j+k_1+1-k_1) \bmod (k_1+1)]+1$

Step 3.3.2: for pre=1…(k_1–1) do

Step 3.3.2.1: $q=(j+k_1+1-pre) \bmod (k_1+1)$

Step 3.3.2.2: for p=0,…,$2^{H[q]}$–1 do //p是一个在$curColum$-pre列上的划分函数

Step 3.3.2.2.1: if $K[j,P]>K[q,p]$ continue;

//判断列 $curColum$ 的划分函数 P 和列($curColumn$-pre)的划分函数 p 是否兼容

Step 3.3.2.2.2: else if $Compatible(curColum,curColumn\text{-}pre,P,p)$

Step 3.3.2.2.2.1: $K[j,P]=\max(K[j,P],K[q,p]+1)$;

Step 3.3.2.2.2.2: $A[j]=\max(A[j], K[j,P])$; //目前看到的最大值

Step 4: $\boldsymbol{M}$ 的 MSR 解就是 $A[i]$

从式(2)可以看出,计算$K[j,P]$只需知道$A[j-k_1]$以及列j前k_1-1 列的相关K值(见$OK(j,P)$的定义),而更新$K[j,P]$

时就可以同时检查$A[j]$是否需要更新,所以,算法需要保存的值是:$A[j-k_1]$到$A[j]$共(k_1+1)个单元,从列$j-k_1+1$到j需要保存k_1列的相关K值,虽然实际程序中K比A可以少 1 个单元,但是为了叙述简便,算法中A和K的值都采用长度为k_1+1 的循环队列来存储,以减小空间需求.算法中需要调用的判断在两个列上的划分函数是否兼容的*Compatible*子过程如下:

```
Boolean Compatible(column1,column2,P,P')
//判断列 column1 上的划分函数 P 和列 column2 上的划分函数 P'是否兼容
{   i=rowset(column1)和 rowset(column2)共有的第 1 行(如果没有共有的行,则 i=–1).
    same=–1;
    while (i!=–1) do
    {  Case1:  M_{i,Column1}!='–' AND M_{i,Column2}!='–'
                   if (same==–1) then same=(M_{i,Column1}==M_{i,Column2})
                   else if (same!=(M_{i,Column1}==M_{i,Column2})) return FALSE;
       Case2:  M_{i,Column1}=='–' AND M_{i,Column2}!='–'
                   Char1=(P 的最低位==1?'B': 'A'); P>>1;
                   if (same==–1) then same=(Char1==M_{i,Column2})
                   else if (same!=(Char1==M_{i,Column2})) return FALSE;
       Case3:  M_{i,Column1}!='–' AND M_{i,Column2}=='–'
                   Char2=(P'的最低位==1?'B': 'A'); P'>>1;
                   if (same== –1) then same=(M_{i,Column1}==Char2)
                   else if (same!=(M_{i,Column1}==Char2)) return FALSE;
       Case4:  M_{i,Column1}=='–' AND M_{i,Column2}=='–'
                   Char1=(P 的最低位==1?'B': 'A'); P>>1;
                   Char2=(P'的最低位==1?'B': 'A'); P'>>1;
                   if (same==–1) then same=(Char1==Char2)
                   else if (same!=(Char1==Char2)) return FALSE;
       i=rowset(column1)和 rowset(column2)共有的下一行(如果没有了,则 i=–1);
       }
       return TRUE;
}
```

定理 3. 对于一个$m\times n$满足(k_1,k_2,h)参数化条件的SNP矩阵$\boldsymbol{M}$,PM_MSR算法能够正确地求出其PM_MSR问题的一个解;该算法的时间复杂度为$O((n-1)(k_1-1)k_2 2^{2h}+(k_1+1)2^h+mn)$,空间复杂度为$O((k_1+1)(n+1)2^h+mn+nk_2)$;如果$\boldsymbol{M}$采用压缩的存储方式,则时间复杂度为$O((n-1)(k_1-1)k_2 2^{2h}+(k_1+1)2^h+nk_2+mk_1)$,空间复杂度为$O((k_1+1)(n+1)2^h+nk_2+mk_1)$.

证明:首先分析其时间复杂度:Step 1 为$O(mn)$;Step 2 为$O((k_1+1)2^h)$;Step 3 执行$n-1$次,Step 3.2 可以通过扫描覆盖*rowset*(*curColum*)中的行在列*curColum*的值得到,总的时间为$O((n-1)k_2)$,由于$H[j]$的值不会超过h,所以, Step 3.3 最多循环$(n-1)2^h$次,Step 3.3.2 最多循环$(n-1)(k_1-1)2^h$次.因为,*rowset*(*column*1)和*rowset*(*column*2)共有的行数不会超过k_2,所以,判断两划分函数是否兼容的子过程*Compatible*所需时间为$O(k_2)$.这样,Step 3.3.2.2 的时间为$O((n-1)(k_1-1)k_2 2^{2h})$.因此,加上预处理PM_MSR算法的时间复杂度为$O((n-1)(k_1-1)k_2 2^{2h}+(k_1+1)2^h+mn)$.下面再进行空间复杂度分析:数组$A$和$H$所需的空间为$O(k_1+1)$,$K$所需的空间为$O((k_1+1)2^h)$,如果还保留对应的*Column*$[j,P]$,则要$O((k_1+1)n2^h)$的空间.*rowset*所需的空间为$nk_2$,加上SNP矩阵$\boldsymbol{M}$,总的空间复杂度为$O((k_1+1)(n+1)2^h+mn+nk_2)$.如果$\boldsymbol{M}$采用压缩的存储形式,则Step 1所需的时间为$O(mk_1)$,预处理时间为$O(nk_2+mk_1)$,所以,加上预处理PM_MSR算法的时间复杂度为$O((n-1)(k_1-1)k_2 2^{2h}+(k_1+1)2^h+nk_2+mk_1)$,空间复杂度为

$O((k_1+1)(n+1)2^h+nk_2+mk_1)$.

PM_MSR 算法的正确性证明:PM_MSR 算法的正确性由初始值式(1)和递推式(2)的正确性来保证.

显然,初始值 $K[1,P]=1$ 是正确的,下面利用数学归纳法来证明递推式(2)的正确性.

假设对于任意 $j'<j$ 和在列 j' 上的任意划分函数 P',$K[j',P']$都是正确的,现在来证明递推式(2)能够正确地求出 $K[j,P]$.

对于列j前的某一列$j'>j-k_1$,如果该列存在与P兼容的划分P',即$(j',P')\in OK(j,P)$,则根据定义 5 中的条件(2),对$\boldsymbol{M}[:,j']$中所有的行存在着一个划分,使得划分在同一子集中的行在$Column[j',P']$中的列上均不发生冲突,且划分在同一子集中的行,如果其覆盖列j',则其P'函数值必相同,再根据定义 6,这些行如果同时也覆盖了列j,则其P函数值必相等,这样根据定义 4,这些行在列j上也不会发生冲突;而那些没有覆盖列j的行,它们自然不会在列j和任何其他行发生冲突.如果只保留$Column[j',P']$中的列和列j,对于$\boldsymbol{M}[:,j]$而言,还需要考虑的只有$\boldsymbol{M}[:,j]$中覆盖了列j但却没有覆盖列j'的行,也就是说,这些在$Column[j',P']$上的取值为空的行(注意:列j是$Column[j',P']$中的最后一列),这些行则可以按照其P函数值分配到具有相同P函数值的其他同时覆盖列j和j'的行所在的子集中,如果没有同时覆盖列j和j'的行,则按P函数值的不同分配到不同的子集中.至于其他行,由于不覆盖$Column[j',P']$中的列,也不覆盖列j,它们在$Column[j',P']$中的列和列j上肯定不会与其他行发生冲突,可以任意分配.这样就对$\boldsymbol{M}[:,j]$中的所有行存在着一个划分,使得划分在同一子集中的行在$Column[j',P']$的列及列j上均不发生冲突,满足定义 5 中的条件(2),由于列j保留,定义 5 中的条件(1)也满足,所以有$Column[j,P]$至少比$Column[j',P']$多一列,这一列就是列j,因此有$K[j,P]\geq K[j',P']+1$.

如果列$j'\leq j-k_1$,由于SNP矩阵$\boldsymbol{M}$满足(k_1,k_2,h)参数化条件,所以没有一行覆盖的列数可以超过k_1,也就是说,没有一行可以同时覆盖列j、列j'或列j'前面的列,否则该行覆盖的列数将超过k_1.对列j'上任意分组函数P'而言,因为列j'是$Column[j',P']$中的最后一列,所以覆盖$Column[j',P']$中任意列的行都不会覆盖列j,它们在列j上的值应为空,按照定义 5,这些行总存在一个划分,使得同一子集中的行在$Column[j',P']$中的列上不会发生冲突,由于它们在列j上的值都为空,所以它们也不会在列j上发生冲突;与$\boldsymbol{M}[:,j']$比较,$M[:,j]$中需要多考虑的行是那些覆盖列j的行,这些行在$Column[j',P']$中所有列上的取值必定为空,所以它们不会与其他行在$Column[j',P']$中的列上发生冲突,按照P函数值对这些行进行划分,值为‘A’的加入到其中一个子集中,值为‘B’的加入到另一个子集中.根据P函数的定义,加入在同一子集中的行在列j上是不会冲突的;而其他既不覆盖$Column[j',P']$中的列,也不覆盖列j的行,它们在$Column[j',P']$中的列及列j上不会与任何其他行发生冲突,对它们进行任意分配.这样,在$\boldsymbol{M}[:,j]$的所有行上存在一个划分,使得划分在同一子集中的行在$Column[j',P']$的列及列j上均不发生冲突,满足定义 5 的条件(2),由于列j保留,定义 5 的条件(1)也满足,所以对任意$j'\leq j-k_1$及其上的任意分组函数P',有$K[j,P]\geq K[j',P']+1$,即$K[j,P]\geq A[j-k_1]+1$.

这样就说明$K[j,P]$至少要比$A[j-k_1]$和$K[j',P']$大 1(对所有的$(j',P')\in OK(i,P)$).

反过来,对于任意列j及其上的划分函数P来说,按照定义 5,对$\boldsymbol{M}[:,j]$中的所有行存在一个划分F,使得划分在同一子集中的行在$Column[j,P]$中的所有列上均不发生冲突,且划分在同一子集中的行,如果其覆盖列j,则其P函数值必相同.那么在同一子集中的行必定在$Column[j,P]-\{j\}$中的所有列上均不发生冲突,即在$Column[j,P]-\{j\}$中的同一列上取‘A’的行和取‘B’的行不会出现在同一个子集里(否则它们就在该列发生冲突).如果$Column[j,P]$中只有列j,那么$K[j,P]=1$,根据等式(3),任意$A[j-k_1]$都不比 0 小,所以有$A[j-k_1]\geq K[j,P]-1$;如果$Column[j,P]$中除列j外还有其他列,则令$Column[j,P]$中除列j外最后一列为j',构造列j'上的划分函数P':由于$\boldsymbol{M}$是一个精简的SNP矩阵,所以在列j'上必有一些行取‘A’值,还有一些行取‘B’值,在划分F中,这些在列j'上取‘A’值的行必定在其中同一个子集中,令该子集中覆盖列j'的所有行的P'值为‘A’;那些在列j'上取‘B’值的行必定在另一个子集中,令该子集中覆盖列j'的所有行的P'值为‘B’.显然,保留$Column[j,P]-\{j\}$中的列能使$\boldsymbol{M}[:,j']$可行,并且存在分区F满足定义 5 的条件(2),条件(1)显然满足,这样就有$K[j',P']\geq K[j,P]-1$,如果$j'>j-k_1$,根据P'的构造方法可知$(j',P')\in OK(j,P)$;如果$j'\leq j-k_1$,则$K[j',P']\leq A[j-k_1]$,就有$A[j-k_1]\geq K[j,P]-1$.

这样就说明,不是$A[j-k_1]\geq K[j,P]-1$ 成立,就是存在一个$(j',P')\in OK(j,P)$使得$K[j',P']\geq K[j,P]-1$ 成立.

递推式(2)成立,定理得证. □

3 实验结果

由于原始的DNA测序片段数据很难得到,文献[15]利用计算机生成的模拟DNA片断数据来比较各片断组装算法的性能,文献[16–18]也是利用计算机模拟真实生物数据的特征生成测试数据集,进行个体单体型各种算法的实验测试,本文也采用与上述文献相同的方法和参数来生成测试数据.模拟数据生成的方法是: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随机生成另一个单体型;然后根据指定的测序误差、片段的覆盖率、片段的最小长度和最大长度来随机生成片段数据集.实验室的DNA测序误差为 3%~5%[17],片段的覆盖率为 5 左右[7,9,12].为了使模拟生成的片段数据能够很好地反映真实情况,根据文献[17]的测试方法,我们先采用著名的shotgun测序模拟片断生成器Celsim[19]生成一系列的片断数据,生成参数设置为两个单体型的差异率为 10%,测序误差为 5%,生成的普通片段的最小长度为 3,最大长度为 6,生成的Mate-Pair是由相隔指定SNP位点的来自于同一个单体型长度为 6 的一对普通片段构成,普通片段中生成洞的概率是 1%,单体型长度和片段覆盖率则按照需要进行变化.模拟片断生成器的详细情况参见文献[17,19].

我们用 C++语言在文献[17]所提供的源程序 Fasthare 的基础上实现了 Bafna 的算法 G_MSR 和本文的算法 PM_MSR,在一台有 4 个 Intel Xeon 3.6G CPU、内存为 4G 的安装了 Linux 操作系统的服务器上,通过一系列的模拟数据对这两种算法进行了测试.图 2~图 7 的每一个点均为 100 次重复测试的平均值,图 8 为 20 次重复测试的平均值,

图 2 的结果显示,当片断覆盖率为 10(普通片段的覆盖率为 6、Mate-Pair 的覆盖率为 4)、当 Mate-Pair 两片段的间距为 10 个 SNP 位点时,G_MSR 和 PM_MFR 的运行时间与测试的 SNP 位点数多少的关系.在 SNP 位点数为 200 时,G_MSR 和 PM_MSR 算法的运行时间分别是 1.637 5s 和 0.00 1s;位点数增加到 600 时,G_MSR 达到 71.397 499s,PM_MSR 用了 0.003 5s.这是因为当SNP位点数增加,覆盖率不变时,片断数也随之增加,在算法复杂度分析上,G_MSR 运行的时间增长速度应该是位点数增长速度的 3 次方,而 PM_MSR 的时间则基本呈线性增长.

图 3 则是当片断覆盖率为 10(普通片段的覆盖率为 6、Mate-Pair 的覆盖率为 4)、SNP 位点数为 400、Mate-Pair 两片段的间距 d 从 1 个~13 个 SNP 位点发生变化时,两算法的运行时间曲线.片断中洞的最大数 k 随着 d 的增加而增加,当 d=1 时,G_MSR 算法需 0.328 5s,PM_MSR 算法需 0.000 5s;当 d 增加到 13 时,G_MSR 和 PM_MSR 的运行时间分别是 65.776 497s 和 0.003 5s.这次实验的结果说明,G_MSR 对 k 的大小非常敏感,当 d 增加时,k 也随之增加,其运行时间急剧上升;但 PM_MSR 的时间变化却不大,这说明 PM_MSR 对 k 不敏感.

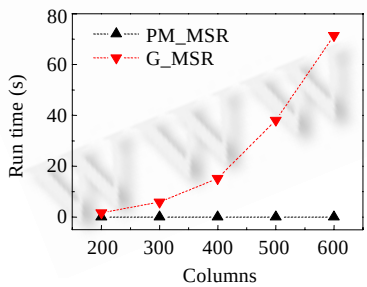

Fig.2 Running time comparison of the algorithms when the number of columns in SNP metrics increases

图 2 SNP 矩阵的列数变化时算法的运行时间对比

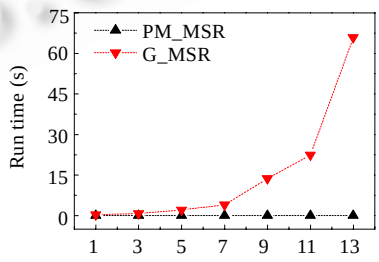

Fig.3 Running time comparison of the algorithms when distance between two fragments of a Mate-Pair increases

图 3 当 Mate-Pair 两片断的间距发生变化时算法的运行时间对比

图 4 是当普通片段的覆盖率为 6、Mate-Pair 中两片段间的间距为 10、SNP 位点数为 400、Mate-Pair 的覆盖率从 2~18 发生变化时(即片断覆盖率从 10 增长到 24),两算法的运行时间曲线图.当 Mate-Pair 的覆盖率为 2 时,G_MSR 和 PM_MSR 算法的运行时间分别是 5.391 5s 和 0.001 5s;当 Mate-Pair 的覆盖率增大到 18 时,两算法的运行时间分别增大到 286.44s 和 49.57s.这个实验说明了 PM_MSR 算法对 Mate-Pair 的覆盖率敏感,这是因为 Mate-Pair 的覆盖率上升时,覆盖同一 SNP 位点且在该位点取空值的片断数就会增加,导致 h 上升,PM_MSR 算法的时间呈指数上升.从实验结果来看,即使在 Mate-Pair 的覆盖率上升到 18 时,PM_MSR 算法所需的时间比 G_MSR 还是要少得多.

当然,如果Mate-Pair的覆盖率继续增大,那么在覆盖某一位点的片断中,在该位点取空值的片断也会增加.当$h>k$时,从算法的复杂度分析可以看出,PM_MSR的性能肯定不如G_MSR,所幸的是,在做DNA测序时,出于成本和时间上的考虑,片段的覆盖率不会很大(人类基因组测序中,覆盖度约为 5[7,9]).

图 5 是当SNP位点数为 600 片断覆盖率为 20(普通片段的覆盖率为 6、Mate-Pair的覆盖率为 14)、Mate-Pair 中两片段间的间距d从 4 减少到 1 时,两算法的对比.当d=4 时,G_MSR和PM_MSR算法的运行时间分别是 25.12s 和 0.3s;当d=1 时,两算法的运行时间分别为 3.91s和 0.02s.从实验结果可以看出,当片断覆盖率保持不变时,d减少时,G_MSR算法和PM_MSR算法的时间都会急剧减少,这是因为d减少时,片段中洞的最大个数k会随之减少,同时,覆盖某位点的片段中在该位点取空值的概率也会着随之变小,即h的值也会跟着变小.极端情况下,片段中没有洞,那么k为 0,h也必定为 0,这时,即使$k_1=n,k_2=m$,根据算法复杂度分析,PM_MSR算法也不会比G_MSR性能明显差.

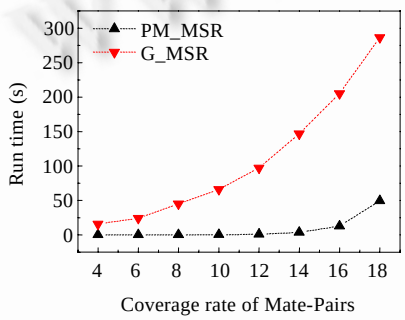

Fig.4 Running time comparison of the algorithms when the fragment coverage of Mate-Pair increases

图 4 Mate-Pair 的覆盖率发生变化时算法的运行时间对比

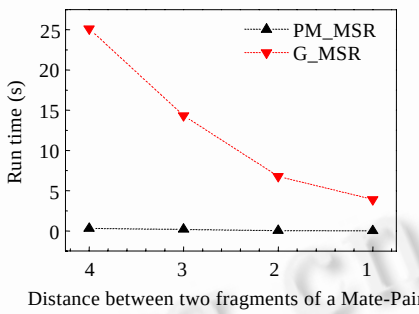

Fig.5 Running time comparison of the algorithms with shorter distance between two fragments of a Mate-Pair

图 5 Mate-Pair 的间距较小时算法的运行时间对比

当 Mate-Pair 的两片断的间距为 100 时,这时片段中洞的最大数 k 可达到 100,G_MSR 已经无法运行了;而 PM_MSR 算法则可以有效地运行.图 6 是当 SNP 位点数为 1 000、Mate-Pair 中两片段间的间距 d 为 100、普通片段的覆盖率为 6、Mate-Pair 的覆盖率从 6 增长到 14 时(即总的片段覆盖率从 12 增长到 20),PM_MSR 算法的运行时间曲线.当 Mate-Pair 的覆盖率为 6 时, PM_MSR 算法的运行时间为 0.284s;当 Mate-Pair 的覆盖率增大到 14 时,运行时间增大为 14 414.125s,约 4h.当 Mate-Pair 的覆盖率增加到 16 时,我们对 PM_MSR 进行了一次测试,时间为 231 913s,约 65h.从实验结果可以看出,当 Mate-Pair 的两片断的间距很大(k 也会很大)而片断覆盖率不大时,PM_MSR 算法能够有效运行;但 h 会随着 Mate-Pair 的覆盖率的增加而增加,从而导致 PM_MSR 算法的时间呈指数上升.幸运的是,为了减少测序的时间和金钱的代价,实验室进行测序时的片断覆盖率是不太大的.

图 7 是当SNP位点数为 600、片断覆盖率为 14(普通片段的覆盖率为 8、Mate-Pair的覆盖率为 6)、Mate-Pair 中两片段间的间距d=10 时,通过控制普通片断的长度改变k_1的值(为Mate-Pair覆盖的SNP位点数)来测试 PM_MSR算法的运行性能.当k_1=18 时,PM_MSR算法的运行时间为 0.0275s;随着k_1增加算法的时间逐渐减少,当

k_1增大为 42 时,算法的运行时间减少到 0.007 5s.这是因为在SNP位点数和片断覆盖率保持不变的情况下,片段越长,片断数越少;又因为构成Mate-Pair的两片段中洞的个数保持不变(为d),这样,所有片断中洞所占的比例就会下降,覆盖同一SNP位点的片断中在该位点取空值的片断数的比例就会下降,进而导致h下降.基于以上两个因素,算法的运行时间将会减少.

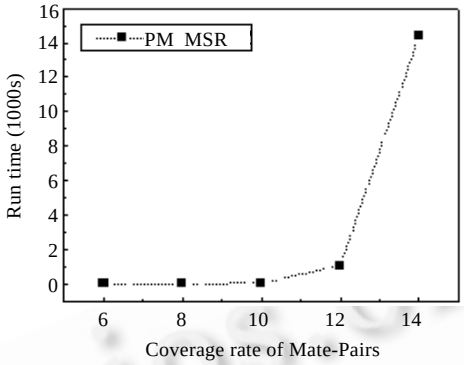

Fig.6 Running time of PM_MSR with 100 SNPs between two fragments of a Mate-Pair

图 6 Mate-Pair 两片断的间距为 100 个 SNP 位点时 PM_MSR 算法的运行时间

图 8 是当SNP位点数为 600、Mate-Pair中两片段间的间距d=10、Mate-Pair的覆盖率为 6,通过控制普通片断的覆盖率改变k_2的值(等于Mate-Pair和普通片断的覆盖率之和)来测试PM_MSR算法的运行性能.当k_2=12时,PM_MSR算法的运行时间为 0.01s;当k_2增大为 24 时,算法的运行时间增大为 0.06 2 秒.从图 8 可以看出,PM_MSR算法运行时间基本上与k_2呈线性增长关系,这与算法复杂度的理论分析结果是相吻合的.

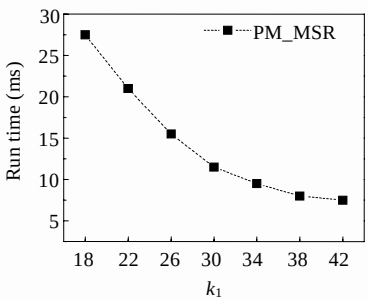

Fig.7 Running time of PM_MSR when k_1 increases

图 7 k_1变化时PM_MSR算法的运行时间

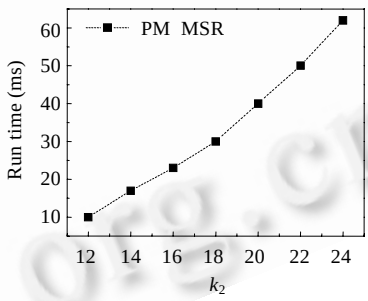

Fig.8 Running time of PM_MSR when k_2 increases

图 8 k_2变化时PM_MSR算法的运行时间

4 结 论

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,而MSR是利用个体基因组片段测序数据用计算方法推出个体单体型的一种优化模型.在DNA测序中,由于碱基的可信度达不到某个给定的阈值或者在采用双管测序法(double-barreled)得到Mate-Pair数据时,片段数据中就会出现空隙.Mate-Pair的典型长度有 2KB,10KB,50KB和150KB[12],按1KB DNA片段中有1个SNP位点[7−9]估计,典型的Mate-Pair片断中洞的个数是2,10,50,150个.这样,当测序数据中有长为 50KB以上的Mate-Pair片段时,片段中洞的最大个数k可达到 50 以上,文献[6]的求解MSR问题的时间复杂度为$O(2^k n^2 m)$、空间复杂度为$O(2^k n^2)$(m是总片段数,n为SNP位点数)的算法无疑是不切实际的.由于受DNA测序成本和时间的限制,基因组的测序数据中的片断覆盖率不会很大,因而覆盖同一个SNP位点的片段数是有限的(约 5 左右,通常小于 20[7,9,12]),令其最大值为k_2,那么,其中在该位点上取空值的片段数不会比k_2大,加之单一片断覆盖的SNP位点一般小于要探测的SNP位点总数.根据以上事实,本文提出了新的参数化

算法.当DNA片断数为m、每个片段覆盖的SNP位点不超过k_1、要确定的单体型的SNP位点数为n、覆盖同一SNP位点的片段数不超过k_2、覆盖某个位点且在该位点取空值的片段数不超过h的情况下,本文提出的时间复杂度为$O((n-1)(k_1-1)k_2 2^{2h}+(k_1+1)2^h+nk_2+mk_1)$、空间复杂度为$O((k_1+1)(n+1)2^h+nk_2+mk_1)$的算法使求解有Mate-Pair片断数据的MSR问题的时间复杂度显著降低,效率提高了很多.该算法的优良特性就是,算法的时间复杂度不再与片段中洞的最大数直接相关,只与覆盖某一位点的片段中在该位点取空值的片段数有关,因而,即使片段数据中有长度为 150KB的Mate-Pair,在片段的覆盖率<20 时,PM_MSR算法也能在较短的时间内完成计算,具有很好的可扩展性.

References:

[1] Miller PT, Gu Z, Li Q, Hillier L, Kwok PY. Overlapping genomic sequences: A treasure trove of single-nucleotide polymorphisms. Genome Research, 1998,8(7):748−754.

[2] Stephens JC, Schneider JA, Tanguay DA, Choi J, Acharya T, Stanley SE, Jiang R, Messer CJ, Chew A, Han JH, Duan J, Carr JL, Lee MS, Koshy B, Kumar AM, Zhang G, Newell WR, Windemuth A, Xu C, Kalbfleisch TS, Shaner SL, Arnold K, Schulz V, Drysdale CM, Nandabalan K, Judson RS, Ruano G, Vovis GF. Haplotype variation and linkage disequilibrium in 313 human genes. Science, 2001,293(5529):489−493.

[3] Horikawa Y, Oda N, Cox NJ, Li X, Melander MO, Hara M, Hinokio Y, Lindner TH, Mashima H, Schwarz PEH, Plata LB, Horikawa Y, Oda Y, Yoshiuchi I, Colilla S, Polonsky KS, Wei S, Concannon P, Iwasaki N, Schulze J, Baier LJ, Bogardus C, Groop L, Boerwinkle E, Hanis CL, Bell GI. Genetic variation in the gene encoding calpain-10 is associated with type 2 diabetes mellitus. Natuture Genetics, 2000,26(2):163−175.

[4] Lancia G, Bafna V, Istrail S, Lippert R, Schwartz R. SNPs problems, complexity and algorithms. In: Heide FM, ed. Proc. of the 9th Ann. European Symp. on Algorithms. LNCS 2161, Heidelberg: Springer-Verlag, 2001. 182−193.

[5] Roach JC. Random subcloning, pairwise end sequencing, and the molecular evolution of the vertebrate trypsinogens [Ph.D. Thesis]. Seattle: University of Washington, 1998.

[6] 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.

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

[8] The Int'l SNP Map Working Group. A map of human genome sequence variation containing 1.42 million single nucleotide polymorphisms. Nature, 2001,409(6822):928−933.

[9] Venter JC, Adams MD, Myers EW, *et al*. The sequence of the human genome. Science, 2001,291(5507):1304−1351.

[10] The Int'l HapMap Consortium. A haplotype map of the human genome. Nature, 2005,437(7063):1299−1320.

[11] Sanger F, Nicklen S, Coulson AR. DNA sequencing with chain-terminating inhibitors. Proc. of the National Academy of Sciences, 1977,74(12): 5463−5467.

[12] Huson DH, Halpern AL, Lai Z, Myers EW, Reinert K, Sutton GG. Comparing assemblies using fragments and Mate-Pairs. In: Gascuel O, Moret BME, eds. Proc. of the 1st Int'l Workshop on Algorithms in Bioinformatics. LNCS 2149, Heidelberg: Springer-Verlag, 2001. 294−306.

[13] Hinds DA, Stuve LL, Nilsen GB, Halperin E, Eskin E, Ballinger DB, Frazer KA, Cox DR. Whole-Genome patterns of common DNA variation in three human populations. Science, 2005,307(5712):1072−1079.

[14] Gabriel SB, Schaffner SF, Nguyen H, Moore JM, Roy J, Blumenstiel B, Higgins J, DeFelice M, Lochner A, Faggart M, Cordero SNL, Rotimi C, Adeyemo A, Cooper R, Ward R, Lander ES, Daly MJ, Altshuler D. The structure of haplotype blocks in the human genome. Science, 2002,296(5576):2225−2229.

[15] Li L, Khuri S. A comparison of DNA fragment assembly algorithms. In: Valafar F, Valafar H, eds. Proc. of the Int'l Conf. on Mathematics and Engineering Techniques in Medicine and Biological Sciences. Nevada: CSREA Press, 2004. 329−335.

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

[17] Panconesi A, Sozio M. Fast hare: A fast heuristic for single individual snp haplotype reconstruction. In: Jonassen I, Kim J, eds. Proc. of the 4th Int'l Workshop on Algorithms in Bioinformatics. LNCS 3240, Heidelberg: Springer-Verlag, 2004. 266−277.

[18] Hüffner F. Algorithm engineering for optimal graph bipartization. In: Nikoletseas SE, ed. Proc. of the 4th Int'l Workshop on Experimental and Efficient Algorithms. LNCS 3503, Heidelberg: Springer-Verlag, 2005. 240−252.

[19] Myers G. A dataset generator for whole genome shotgun sequencing. In: Lengauer T, Schneider R, Bork P, Brutlag DL, Glasgow JI, Mewes HW, Zimmer R, eds. Proc. of the 7th Int'l Conf. Intelligent Systems for Molecular Biology. AAAI Press, 1999. 202–210.

谢民主(1969－),男,湖南涟源人,博士生,主要研究领域为生物信息学.

王建新(1969－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为计算机网络,网络优化理论,计算机优化算法,生物信息学.

陈建二(1954－),男,博士,教授,博士生导师,主要研究领域为生物信息学,计算机理论,计算复杂性及优化,计算机网络优化算法.