分类号 TP301　　密级

U D C　　编号

中　南　大　学

博 士 后 研 究 工 作 报 告

单体型计算与全基因组关联分析算法研究

谢 民 主

工作完成日期　2008 年 12 月—2011 年 12 月

报告提交日期　2011 年 12 月

中　南　大　学　（湖南）

2011 年 12 月

单体型计算与全基因组关联分析算法研究

THE EXPLORATION FOR NEW ORGANIC NLO —MATERIALS TIIE MATERIALS CIIEMISTRY PROCESS —FROM MOLECULES TO CRYSTALS

博　士　后　姓　名　谢 民 主

流动站（一级学科）名称　控制科学与工程

专　业（二级学科）名称　模式识别与智能系统

研究工作起始时间　2008 年 12 月 24 日

研究工作期满时间　2011 年 12 月 20 日

中南大学（湖南）

2011 年 12 月

内 容 摘 要

本文对单体型组装、人类白细胞抗原(HLA)等位基因推断和全基因组关联分析(GWAS)进行了深入研究。

分析和识别单体型对复杂疾病致病基因的精确定位有重要作用，单体型组装问题是利用个体 DNA 测序片段数据推出该个体一对单体型的计算问题。在实际 DNA 片段数据中，一个片段所覆盖的最大 SNP 位点数 k 通常小于 10，据此本文对最小错误纠正模型（MEC）提出了一个新的时间复杂度为 $O(mk2^k+m\log m+mk)$、空间复杂度为 $O(mk2^k)$的参数化动态规划精确算法。由于测序实验中 k 通常很小，该算法实用高效。由于最优解可能有多个，本文对 MEC 的一个扩展模型 MEC/GI 设计了能提供 top-k 个优化解的遗传算法，大量模拟实验显示多个优化解中包含真实解的概率比一个最优解就是真实解的概率要高。

HLA 基因在人类免疫系统中有重要作用，HLA 基因不匹配是器官移植失败的一个重要原因，HLA 基因的不同变异与许多免疫疾病、炎症和感染有关联。用血清学和 PCR 等生物实验测试方法直接确定 HLA 基因变异耗时耗力，用有效的计算技术帮助确定 HLA 基因有重要意义。本文基于加权单体型相似图设计了一个 HLA 基因推断算法。在一组预先测定了 SNP 基因型和 HLA 基因的数据集上的大量实验测试显示该算法能根据 HLA 基因相邻区域的 SNP 基因型数据精确地进行 HLA 基因推断。

利用大量个体全基因组 SNP 位点上基因型信息和相关疾病表型信息进行全基因组关联分析(GWAS)是揭示复杂疾病致病基因的有效手段，由于受到计算资源的限制，目前 GWAS 采用的主要模式是疾病与单个 SNP 位点相关统计分析的方法，可是人类复杂疾病往往是多个基因的交互作用的结果。本文基于相对频繁项聚类设计了一个简单、快速而有效的全基因组多基因交互探测算法，该算法把所有 d-位点的基因型组合划分为 3 组，然后用 χ^2 进行统计显著性评估，结合排列测试（pemutation test）和 Bonferroni 纠正进行快速多重测试修正进行假阳性率控制。大量模拟测试显示该算法快速，且比最近提出的一些算法具有更强的探测能力。

关键词：单核苷酸多态性，单体型，基因型，人类白细胞抗原，全基因组关联分析

Abstract

This report presents our research results on three bioinformatics problems: the haplotype assembly problem, the human leukocyte antigen (HLA) allele inference problem, and genome-wide association studies.

Haplotyping plays an important role in locating complex disease susceptibility genes. The haplotype assembly problem is a computational problem that, given a set of DNA sequence fragment data of an individual, induces the corresponding haplotypes. Based on the observation that, for the real DNA sequence fragment data, the maximum number k of SNPs that a fragment covers is usually smaller than 10, we develop a new parameterized dynamic programming algorithm with running time $O(mk2^k+m\log m+mk)$ and space $O(mk2^k)$ for an important model Minimum Errors Correction (MEC), where m is the number of fragments. Since k is small in real biological application, the algorithm is practical and efficient. Since there may be multiple optimal solutions to a problem, we design a genetic algorithm to provide top-k optimal solutions to MEC/GI, an extension of MEC. Extensive experimental results show that it is more probable to find the real solution in top-k solutions when $k > 10$ than when $k = 1$.

The human leukocyte antigen system (HLA) genes play an important role in the human immune system, HLA gene matching is crucial for the success of human organ transplantations, and HLA gene variations are associated with many autoimmune, inammatory and infectious diseases. However, typing HLA genes by serology or PCR is time consuming and expensive, accurate computational algorithms to infer HLA gene types from SNP genotype data are in need. Based on weighted haplotype similarity graphs, we design an accurate HLA gene type inference algorithm. Extensive experimental results on a previously typed dataset have illustrated that the algorithm can infer the HLA gene types from their neighboring SNP genotypes accurately.

Genome-wide association studies (GWAS) have proven to be a powerful approach to reveal susceptibility genes for complex diseases. Limited by the computational resources, the primary analysis paradigm for GWAS is domintated by single-locus based statistical approaches. However, interactions among multiple genes play an essential role in the pathogenesis of human complex diseases. We develop a simple, fast and effective algorithm

to detect genome-wide multi-locus interactions based on the clustering of relatively frequent items. It groups all *d*-locus genotype combinations into three groups and uses the χ^2 statistic to measure significance. To control the false positive error rate, we have combined Bonferroni correction and permutation tests and proposed a fast multi-test correction method. Extensive experiments on simulated data show that our algorithm is fast and more powerful in general than some recently proposed methods.

Keywords: single nucleotide polymorphisms (SNPs), haplotype, genotype, human leukocyte antigen, genome-wide association studies (GWAS)

目　　次

1　生物学背景知识

1.1 研究背景

1990 年 10 月，人类基因组计划（Human Genome Project, HGP）启动，在美、英、日、德、法和中国等国科学家十多年的艰苦努力，2003 年 4 月人类基因组图谱基本完成[1]，至此人类基因组共性的一面被揭示出来。但是不同人有不同的外貌和体格特征，对疾病抵抗能力、对药物的敏感性等均不相同，除了环境等因素外，这主要是因为除了同卵双胞胎外，不同个体的基因组是有差别的。不同人的 DNA 差异约占基因组的 0.5%[2]，单核苷酸多态性（Single Nucleotide Polymorphisms，SNPs）是由单个核苷酸的变异所引起的多态性，一般认为是人类 1%或以上的个体中可以见到的某个位点上的碱基变化[3,4]，在整个人类基因组中有几百万个 SNPs[2, 4, 5]。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。

SNPs 是一个物种中不同个体表型的主要遗传来源，识别 SNPs 对遗传病等疾病的诊断和药物研究有重要作用，亦可用于个体识别、亲子鉴定和人类各群体的遗传关系分析。对于人类等二倍体生物，DNA 的载体——染色体是成对存在。在一条染色体某段连续区域 SNP 位点上的碱基序列叫做单体型（Haplotype）[6]。对于任何一个二倍体生物，都有二个单体型。为构建确定人类遗传的相似性和差异性，由加拿大、中国、日本、尼日利亚、英国和美国共同资助进行的国际人类基因组单体型图计划 HapMap 于 2002 年 10 月正式启动[7]。2005 年 10 月 26 日，HapMap 协作组[1]公布了其初步绘制的人类首张单体型图[6]。2007 年 10 月 18 日，第二期的 310 万个 SNPs 的单体型图已完成[8]。

HapMap 计划极大地推动了单体型的研究和应用，不幸的是在当前的实验技术下，直接使用生物实验测定单体型既费钱又费时间，利用计算机技术来确定个体单体型有重要的现实意义[9, 10]。

人类白细胞抗原(Human Leukocyte Antigen, HLA) 超级基因座(super locus)位于类染色体 6p21 区域，0.5% (> 150 个) 蛋白质编码基因位于 HLA 超级基因座内[11]，且每个基因都有 10 多个不同的变异[12]。HLA 基因在免疫系统里有重要作用，它们编码

[1] http://www.hapmap.org

称为 HLA 复合体（complex）的一组相关蛋白质，人类的免疫系统依靠 HLA 复合体来区分自身细胞和外界入侵细胞。器官移植中供体和受体的 HLA 基因不匹配引起的排斥反应会导致移植的失败。高度的多态性 HLA 区域一直是人类遗传领域的研究热点[13]，很多研究者揭示 HLA 基因变异与许多免疫疾病、炎症和感染有关联[14, 15]，可是用血清学和 PCR 等生物实验测试方法直接确定 HLA 基因变异耗时耗力，制约了有关 HLA 基因的大规模研究[16]，因此特别需要用有效的计算技术帮助确定 HLA 基因。

随着基因分型技术的飞速发展，在最近五年内科研机构收集的常见疾病表型信息及相关个体的全基因组基因型信息呈加速度增长[17-19]。利用大量个体全基因组 SNP 位点上的基因型信息和相关的疾病表型信息进行全基因组关联分析(Genome-wide association studies, GWAS)是揭示复杂疾病致病基因的有效手段[20, 21]。目前 GWAS 采用的主要模式是疾病与单个 SNP 位点相关统计分析的方法[22]，可是人类复杂疾病往往是多个基因的交互作用(epistasis)的结果[23, 24]。大量研究结果显示乳腺癌[25]、糖尿病[23]和冠心病[26]等人类常见疾病与多个基因的交互作用有密切关系。基于单个 SNP 位点的统计方法可能无法探测到所有交互的基因，特别是在交互的多个基因中单个基因变异不能显著影响疾病的发病率的情况下。

本文主要研究三个问题：第一个是如何确定个体的单体型的计算问题，第二个是 HLA 基因推断问题，第三个是 GWAS 多基因交互分析问题。下面对相关的背景知识进行详细的阐述。

1.2 遗传的物质基础及遗传法则

生命缤纷多彩，物种千变万化，但是种豆得豆，种瓜得瓜，是什么在控制生命的诞生和发展的进程？公元前五世纪希波克拉底（Hippocrates）认为子代具有亲代的特性是因为在精液或胚胎里集中了来自身体各部分的微小代表元素（Element）；100 年后，亚里斯多德（Aristotle）认为精液提供了后代的蓝图，生物的遗传是个体胚胎发育所需信息的传递。1865 年奥地利人孟德尔（G. J. Mendel）从他 8 年植物杂交实验的结果发现了生物每一个性状都是通过遗传因子来传递的，遗传因子是一些独立的遗传单位。1909 年，丹麦人约翰森（W. L. Johannsen）将孟德尔提出的“遗传因子”改称为后来广为流传的“基因”（Gene）。1910 年摩尔根（T. H. Morgan）等创立了连锁定律，1944 年艾弗里（O. T. Avery）确定遗传物质为脱氧核糖核酸（Deoxyribonucleic acid，DNA），1953 年沃森（J. D. Watson）和克里克（F. H. C. Crick）建立 DNA

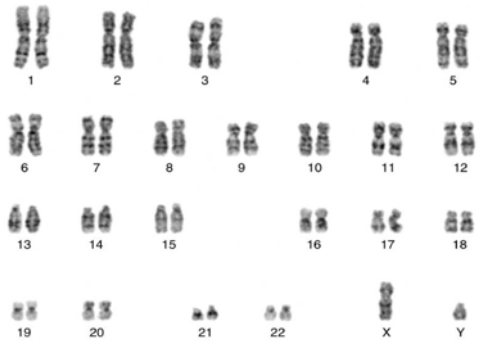


图 1.1 人类男性体细胞染色体

分子的双螺旋结构模型。至此人们普遍认为生命个体的遗传信息包含在细胞中全部 DNA 所构成的基因组中，生命的表型和其他性状是由基因组决定的。

1.2.1 染色体

对于人类这样的真核生物，生命体的基本单位细胞（Cell）中有细胞核（Nucleus），在细胞核中有染色体（Chromosome）。1910 年摩尔根等创立连锁定律的同时也证实了遗传信息（基因）在染色体上以线状排列。

染色体的化学成分主要是 DNA 和蛋白质，在细胞发生有丝分裂时期容易被碱性染料着色，因此而得名。染色体是遗传物质的主要载体，细胞中大部分的 DNA 在染色体上，染色体中 DNA 含量稳定，是主要的遗传物质。

在无性繁殖物种中，生物体内所有细胞的染色体数目都一样。而在有性繁殖物种中，生物体的体细胞染色体成对分布，称为二倍体。人类体细胞中含有 22 对常染色体，1 对性染色体。男性的性染色体对由 X 和 Y 染色体构成（如图 1.1[2]），女性的性染色体由 2 条 X 染色体构成。其中每一对染色体中，一条来自于母亲，另一条来自于父亲。

1.2.2 DNA 分子与基因

构成 DNA 分子的基本单位是脱氧核苷酸（Deoxyribonucleotide），脱氧核苷酸由磷酸（Phosphate）、脱氧核糖（Deoxyribose）和碱基（Base）构成，其中碱基有四种：腺嘌呤（Adenine，缩写为 A）、鸟嘌呤（Guanine，缩写为 G ）、胞嘧啶（Cytosine, 缩写为 C ）和胸腺嘧啶（Thymine, 缩写为 T），由此脱氧核苷酸可分为四种，分别用 A、C、G、T 表示。脱氧核苷酸之间通过 3′、5′-磷酸二酯键相连形成脱氧核苷酸链，两

[2] http://www.imagewa.com/Photo/269/100.html

条脱氧核苷酸链平行但反向盘旋成的规则的 DNA 分子双螺旋结构，两条链之间是通过互补碱基配对连接在一起，其中 A 与 T 以 2 个氢键相配对，C 与 G 之间以 3 个氢键配对。DNA 的双螺旋结构如图 1.2 所示[3]，碱基互补配对如图 1.3 所示[127]。在图 2.4 中，上面是 DNA 分子的化学结构，下面是该分子的碱基对序列。在 DNA 分子的化学结构图中，P 表示磷酸，D 表示脱氧核糖，A、C、G 和 T 分别表示四种碱基，一个 DNA 单链开始于 3′端（脱氧核糖端），中止于 5′端（磷酸端）。DNA 分子的一级结构指 DNA 分子中核苷酸的排列顺序，即核苷酸序列或碱基序列，DNA 分子的多样性是由碱基排列顺序的多样性决定的。

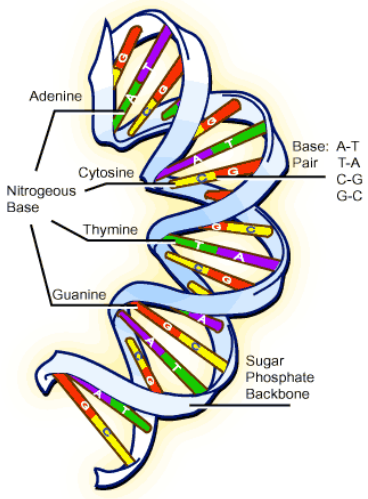


图 1.2 DNA 双螺旋结构

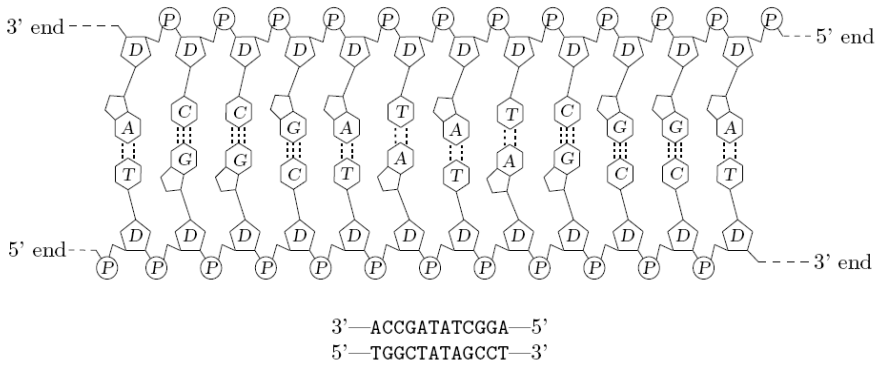


图 1.3 碱基互补配对

现代遗传学认为，基因是指位于染色体的特定位置、编码特异的蛋白质或 RNA

[3] http://www.scq.ubc.ca/?p=263

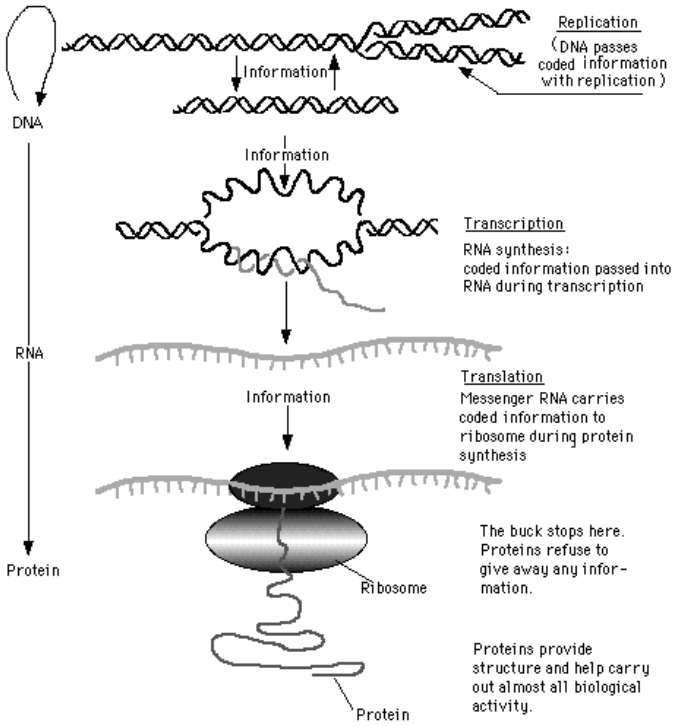


图 1.4 生物信息学的中心法则

的一段核酸序列（通常是 DNA 序列），是遗传物质的结构和功能单位。对于真核生物而言，基因位于染色体上，并在染色体上呈线性排列。基因不仅可以通过复制把遗传信息传递给下一代，还可以使遗传信息得到表达，也就是使遗传信息以一定的方式反映到蛋白质的分子结构上，从而使后代表现出与亲代相似的性状。基因组(Genome)代表了一个生物细胞内的全部基因和染色体组成。

1.2.3 分子生物学的中心法则

1958 年，克里克提出了两个学说，奠定了分子生物学的理论基础。第一个学说是“序列假说”，它认为一段核酸的特殊性完全由它的碱基序列所决定，碱基序列编码一个特定蛋白质的氨基酸序列，蛋白质的氨基酸序列决定了蛋白质的三维结构。第二个学说是“中心法则”，遗传信息只能从核酸传递给核酸，或核酸传递给蛋白质，而不能从蛋白质传递给蛋白质，或从蛋白质传回核酸。后来，沃森把“中心法则”更明确地表示为，遗传信息只能从 DNA 传到 RNA，再由 RNA 传到蛋白质。分子生物学的中心法则（Central dogma of molecular biology）如图 1.4[4]所示，从“中心法则”可

[4] http://www.cbs.dtu.dk/staff/dave/roanoke/genetics980320f.htm#1.%20Digital%20River

以看出，遗传信息的一般流动方向是：遗传信息可以通过 DNA 的自我复制（Replication）从 DNA 流向 DNA，也可以通过转录过程（Transcription）从 DNA 流向 RNA，进而通过翻译过程（Translation）从 RNA 流向蛋白质，最后通过具有空间结构的蛋白质完成细胞的各项生命活动。

1.3 单核苷酸多态性、单体型和基因型

人类不同个体有不同的外貌和体格，对疾病有不同的抵抗能力，从遗传上说是因为不同的人的基因组不完全相同。人类的基因组包含 23 对染色体上的 DNA 序列，总长度达 3 亿个碱基。不同人的 DNA 序列极为相似，若对两个不同的人的同源染色体进行比较，他们的 DNA 序列上可以连续数百个碱基都是相同的。最近的研究[3]表明两个不同的同源染色体的 DNA 序列中，99.5%是相同的，其他不同的部分可能是短的 DNA 片段的插入（Insertion）、删除（Deletion）和单个碱基的差异，其中主要是单个碱基的差异。在同一个物种中，这些遗传物质上的变化如果在群体中发生频率低于 1%则称为变异（Mutation），如果不小于 1%，则称多态现象（Polymorphism）。

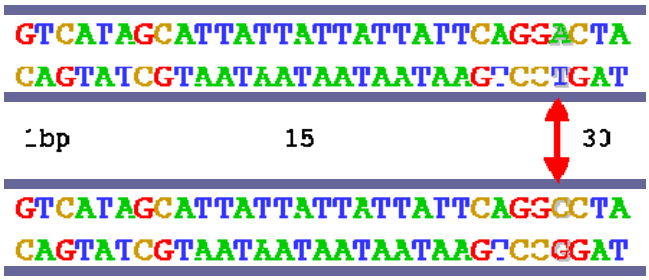


图 1.5 单核苷酸多态性

单核苷酸多态性（Single Nucleotide Polymorphism，SNP）：单核苷酸多态性指染色体基因组上在单个核苷酸碱基尺度上的变化（这种变化最少在群体中的频率不小于 1%）而引起的 DNA 序列多态性。在图 1.5[5]中，第 30 个碱基处是一个 SNP 位点。虽然脱氧核苷酸碱基有四种，但 SNP 通常只是二等位基因（Biallelic）的变异，即只含两种等位基因型（碱基对）。

人类个体内超过 90%的多态现象都是单核苷酸多态性（SNPs）。由于在 DNA 序列中，大约 1000 个碱基中有 1 个 SNP，所以 SNP 作为分子标记广泛地用于致病基因定位等后基因组研究之中。

单体型（Haplotype）：单体型指的是一条染色体上或染色体一段区域内相关的

[5] http://cmbi.bjmu.edu.cn/cmbidata/snp/index00.htm

SNP 序列。在图 1.6 中，对应的三个单体型分别为 CTC、CAT 和 ATC。 确定染色体上的单体型叫单体型分型（Haplotyping）。

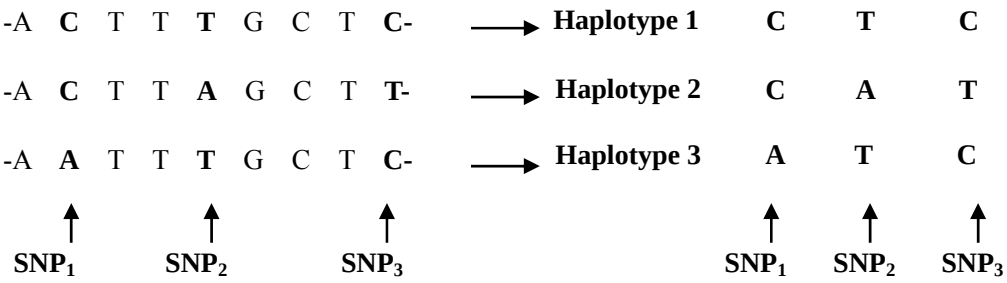


图 1.6 单体型

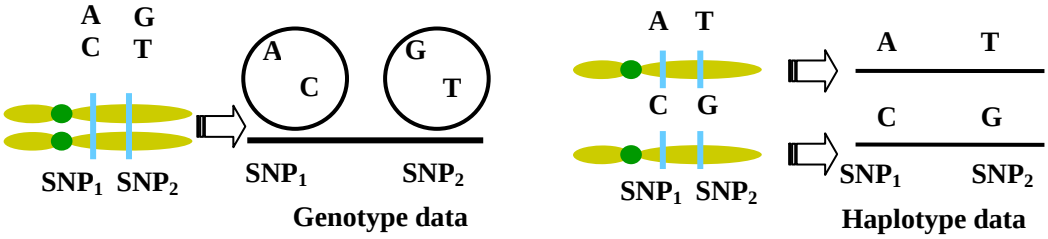


图 1.7 一对染色体的基因型及对应的单体型

基因型（Genotype）：我们常见的大多数动植物都是双倍体（Diploid），人类也是如此。人的细胞核内的染色体成对存在，成对存在的同源染色体其中一条来自父亲，另一条来自母亲。由于在 DNA 测序中很难把同源染色体分离，这样通常实验室里面得到的就是这两条同源染色体共同表现出来的 SNP 复合序列，称之为基因型。在图 1.7 中，左边是某个体的一对同源染色体对应的基因型，表示为 A/C G/T；由左边的基因型信息是无法准确确定右边的两个单体型，因为由 AG 和 CT 这两条单体型构成的基因型也是 A/C G/T。

对于任意一个 SNP 位点来说，一对同源染色体上的碱基可以是相同的，也可以是不同的。同源染色体在某个 SNP 位点上的取相同碱基值的现象叫纯合(homozygous)，这时称这个 SNP 位点上的基因型为纯合子（homozygote）。同源染色体在某个 SNP 位点上的碱基值的不同的现象叫杂合(heterozygous) ，这时称这个 SNP 位点上的基因型为杂合子（heterozygote）。

1.4 人类遗传病

对人类自身认识不断深入的研究越来越表明除了与后天的环境因素有关外，人类

的很多疾病都与家族遗传有关，个体罹患疾病的可能性与个体的遗传物质密切相关。

人类的遗传病可以分为以下三大类。

(1) 单基因病（monogenic disorder）：这种疾病由 DNA 序列的某个碱基对的改变所致，如血友病、白化病、红绿色盲、杜氏肌营养不良、镰刀型细胞贫血症和抗维生素 D 佝偻症等，这种疾病的遗传符合孟德尔遗传定律，因此也叫孟德尔疾病（Mendelian disorder）。对单基因遗传病致病的基因的定位已有成熟的方法，每年都有新发现。

(2) 染色体疾病（chromosome disorder）：可进一步分为常染色体异常和性染色体异常两个类型，是由于染色体在数量或结构上的异常，如先天愚型、Edward 氏综合、先天性卵巢发育不全综合征、先天性睾丸发育不全综合征、超雌综合征、真性两性畸形等等，这类疾病很容易通过染色体检查发现。

(3) 多基因病（polygenic disorder）：这种疾病受两对以上等位基因的控制，一般还受环境等多种复杂因素的影响，也叫作人类复杂疾病（complex disease）。多基因病在人群中比较常见，并且发病率极高，如无脑儿、唇颚裂、原发性高血压、癌症、心脏病、哮喘、先天性幽门狭窄、精神分裂症和糖尿病等等。

在人类复杂疾病致病基因的研究上，研究人员发现研究单基因病的方法不再有效，人们必须从整体上研究多个基因及基因与环境之间的关系。在寻找复杂疾病致病基因的研究上，单体型和全基因组关联分析起着非常重要的作用。

1.4 本文的主要研究内容及结构安排

本文主要研究三个与 SNP 相关的生物信息学问题：（1）单体型组装问题；（2）人类白细胞抗原(Human Leukocyte Antigen, HLA)等位基因推断问题；（3）全基因组关联分析问题(Genome-wide association studies, GWAS)。本论文的具体组织如下。

第 1 章 生物学背景知识：本章对整个论文的研究背景和主要研究内容进行概述并对相关的分子遗传学背景知识进行介绍。

第 2 章 单体型组装算法：本章对单体型组装问题的基本概念和研究现状进行详尽阐述之后，对其中重要的计算模型 MEC 及 MEC/GI 进行了研究。对于 MEC/GI 模型，本章根据生物实验中能直接测序的 DNA 片段覆盖的最大 SNP 位点较小的事实，对 MEC/GI 进行参数化建模，设计出时间复杂度为 $O(mk2^k+mlogm+mk)$、空间复杂度

为 $O(mk2^k)$的精确算法 K-MEC/GI。对于 MEC，本章提出了一个能提供 top-k 个优化解的遗传算法。

第 3 章 HLA 推断算法：本章提出了一个基于单体型加权相似图的 HLA 基因推断算法。给定一些家族的基因型数据和其中多数个体的 HLA 基因数据，该算法为 HLA 推断问题建模和单体型优化选择建立一个统一的框架，能根据一个群体的 SNP 基因型数据和其中多数个体的 HLA 基因信息推断出其余个体的 HLA 基因。

第 4 章 基于相对频繁项聚类探测多基因交互：本章提出了一个基于相对频繁项聚类探测多基因交互的 GWAS 算法。该算法 F 把所有 d-位点的基因型组合划分为 3 组，然后用 χ^2 进行统计显著性评估，然后结合排列测试（pemutation test）和 Bonferroni 纠正的优点提出了一种快速的多重测试修正方法控制 EDCF 的假阳性率。

第 5 章 结论及展望：本章对全文的工作进行总结，并对下一步的研究方向进行展望。

2　单体型组装算法

2.1　引言

单个 SNP 可用于单基因疾病致病基因的定位，也可以用于对疾病的关联分析，但是越来越多的研究表明，使用单体型比使用单个 SNP 在复杂疾病的关联研究上更加有效[27-32]。Clark[28]和 Tachmazidou 等[29]也指出在复杂疾病的关联分析中，单体型更有用。Browning 等[30]指出在疾病易感变异（disease-susceptibility variants）的频度在人群中不大于 5%时，基于单体型的测试比基于单个 SNP 的测试更有效。Morris 等[31]和 Epstein 等[32]也指出，当多个与疾病相关的可疑等位基因（susceptibility alleles）之间缺乏连锁关系时，基于单体型的方法优于基于单个 SNP 的分析方法。正是因为这个原因，近年来有无数学者利用单体型进行复杂疾病的关联研究，取得了大量的研究成果[33-44]。显然在复杂疾病的关联分析和致病基因定位中，单体型起着非常重要的作用[45]。

自从 1990 年 Clark 在 Mol. Biol. Evolut.上发表论文提出用一群个体的基因型去推断出每个个体的单体型的思想[46]以来，国外发达国家有很多学者如 D. Gusfield[10, 48-52]、G. Lancia[9, 52, 53]、V. Bafna[55-57]、Rizzi [58] 、J. Li、T. Jiang[59-61]和 L. Wang, Y. Xu[62]等，国内主要有中科院数学与系统科学研究院的章祥荪教授领导的生物信息学研究中心[63-69]、中国科技大学的张强锋[70-75]及中南大学谢民主[76-80]等在等对单体型分型中的计算问题进行了大量研究, 提出了单体型检测的各种模型和算法，这些模型总的说来主要分为单体型组装和单体型推断两大类[65, 76]。本章研究单体型组装问题（The Haplotype Assembly Problem）。

2.2　单体型组装问题相关定义和研究现状

单体型组装问题（Haplotype Assembly Problem）最早是由 Lancia[9]提出，也叫个体单体型问题（Individual Haplotyping Problem）。对于人类等二倍体生物，染色体是成对存在, 都有二个单体型。在当前的技术条件下直接把一对染色体分开，然后对每一条染色体进行独立测序难度很大，花费的金钱和时间过分昂贵，因此，实验室的测出的 DNA 片段数据是来自于一对染色体，而单体型组装问题就是给定一组来自某对

同源染色体的由 DNA 测序方法得到的 DNA 片段数据，根据片段上的 SNP 值组装出两条单体型。

图 2.1 是一个单体型组装问题的示例[81]，其中图 2.1(a)是来自于某个个体的联配的 DNA 测序片段(fragment)数据；而单体型组装问题只是关心 DNA 片段在 SNP 位点上的取值，图 2.1 (b)则标记出了对应的 SNP 值；图 2.1 (c)根据 SNP 的取值，把所有的 DNA 片段划分为两个集合，使得在同一个集合里的片段在对应的 SNP 位点上的取值均相等，即认为同一个集合里的片段来自于同一条染色体，这样就得出该个体的 1 对单体型为“CCACGAGT”和“GATTATCA”。当 DNA 测序过程没有错误时，单体型组装问题很容易得以解决，但是由于测序过程的错误是不可避免的[9]，因此实验室测得的 DNA 片段数据常常不可能划分为两个集合，使得在同一个集合里的片段在对应的 SNP 位点上的取值均相等，这样单体型组装问题在计算上就变得很困难。

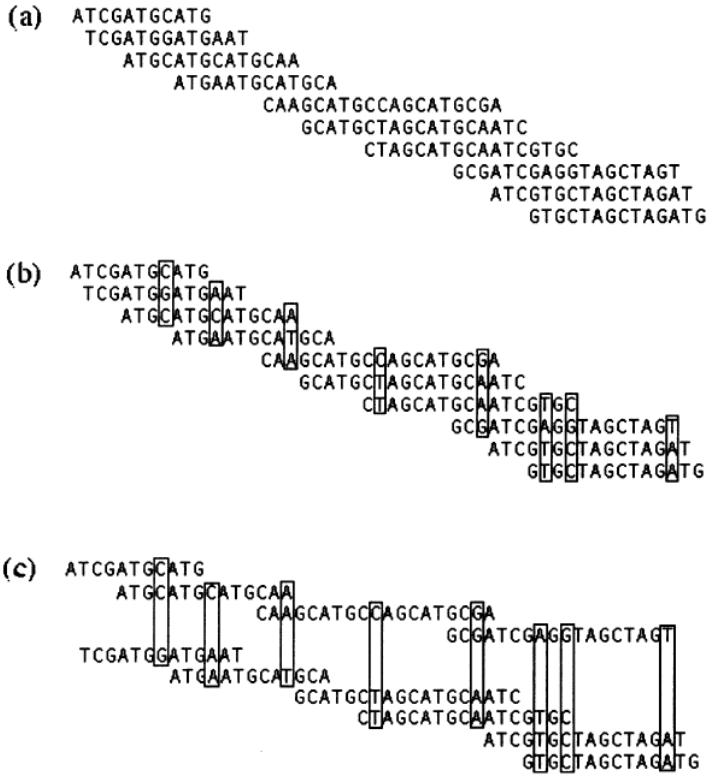


图 2.1 单体型组装问题:（a）个体的联配的 DNA 片段数据；(b)对 SNP 位点做了标记的联配 DNA 片段数据 ；（c）按照 SNP 的取值对 DNA 片段进行的划分

如前所述，单核苷酸多态性 SNPs 指的是在群体 1%以上个体中出现染色体单个位点上的碱基变异。人类基因组包含着广泛分布的几百万个 SNPs。人类等双倍体生物的 DNA 序列是按染色体成对出现的。一条染色体上某一区域的 SNP 位点上的碱基

序列叫做单体型，而一对染色体上对应区域 SNP 位点上碱基对的序列叫做基因型。在图 2.2 中，该个体的单体型是(G, C, A, T, G)和(A, T, A, C, G)；基因型为(A/G, C/T, A/A, C/T, G/G)。

一对染色体对应位点上 SNP 值可以是相同的，这种现象叫纯合(homozygous)；也可以是不同的，这种现象叫做杂合(heterozygous)。这样单体型就可以用字符集{0, 1}上的字符序列来表示, 不必用真正的碱基字符，其中‘0’通常表示人群中大部分单体型在该位点上的 SNP 值, 而‘1’则表示人群中少部分单体型在该位点上的 SNP 值。同理，基因型则可以用字符集{0, 1, 2}上的字符序列来表示，其中 0(或 1)是纯合子，表示两个单体型上对应的 SNP 值都为 0(或 1)，2 是杂合子。图 2.2 中的单体型可表示为 “0 1 0 1 1” 和“1 0 0 0 1” ，其中在第 1 个 SNP 位点‘0’表示‘G’, ‘1’表示‘A’; 在第 2 个 SNP 位点‘0’表示‘T’, ‘1’表示‘C’; 其余的类似。而基因型可表示为 “2 2 0 2 1”。

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

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

图 2.2 单体型

SNP 位点

片段										
	-	-	-	-	-	0	1	-	1	0
	-	0	1	-	-	0	-	-	-	-
	0	1	1	0	-	-	-	-	-	-
	1	0	1	-	0	1	-	-	-	-
	-	1	0	-	-	-	-	-	-	-
	-	-	0	1	-	-	-	-	-	-
	-	-	-	0	1	0	-	-	-	-
	-	-	-	-	-	-	0	1	-	-
	-	-	-	-	-	-	1	0	0	1
	-	-	-	-	0	1	-	-	-	-
	-	-	-	-	-	0	0	-	-	-

图 2.3 SNP 矩阵

由于单体型组装问题只关心 DNA 片段在 SNP 位点上的取值，对于一对染色体某个对应区域内的 *n* 个 SNP 位点按其在染色体上的次序从左到右记作 S：{1，2，...，*n*}，*m* 个 DNA 片段记作 F：{1，2，...，*m*}。任意 SNP 位点应该被某些 DNA 片段覆盖，任意片段在它所覆盖的 SNP 位点的取值为{0，1，−}，其中‘−’为空值，表示该片段没有覆盖该 SNP 位点，或在该位点的取值未知。这样 DNA 片段的数据集可以表示

为在{0，1，−}上的一个 $m\times n$ 的矩阵，叫做 SNP 矩阵 $\mathbf{M}$[17]。图 2.3 是一个 11×10 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，$\mathbf{M}_{i,j}$ 表示第 i 个片段在第 j 个 SNP 位点上的取值。

下面是与 SNP 矩阵 **M** 相关的几个定义。

定义 2.1　如果$(\exists k(\mathbf{M}_{i,k}\neq\text{‘}-\text{’})\wedge(k\le j))\wedge(\exists r(\mathbf{M}_{i,r}\neq\text{‘}-\text{’})\wedge(j\le r)))$，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 2.3 中行 2 覆盖列 2 到 6，但是行 2 没有覆盖列 1，也没有覆盖列 7 到 10。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 2.3 中行 4 在列 4 上有个洞。如果 **M** 的所有行都没有洞，则称 **M** 为**无空隙**的 SNP 矩阵。

M 的行覆盖的最左边和最右边的列号分别用函数 l 和 r 来表示，即行 i 覆盖的最左边的列是 $l(i)$，覆盖的最右边的列是 $r(i)$。在图 2.3 中，$l(2)=2$，$r(2)=6$。

为了叙述简便，本文中把取 **M** 的前 i 行构成的 SNP 矩阵记作 $\mathbf{M}(i,:)$，取 **M** 的前 j 列构成的 SNP 矩阵记作 $\mathbf{M}(:,j)$。

定义 2.2　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**。

如图 2.3 所示的 SNP 矩阵中，行 3 与行 4 在列 1 和 2 上冲突，在其余列上不冲突；行 1 和行 2 在所有的列上均不冲突，它们兼容。

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 2.3　如果 SNP 矩阵 **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 **M** 是**可行**的。

如果 **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。

一个 SNP 矩阵 **M** 是可行的当且仅当可以找到一对单体型，使得 **M** 中的任意行总是可以与其中的一个单体型兼容。这时，称 **M** 可由这对单体型导出。

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

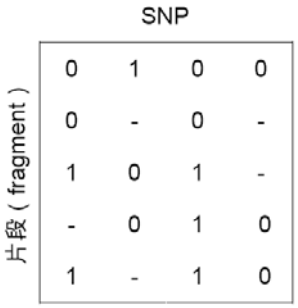


图 2.4 可行的 SNP 矩阵

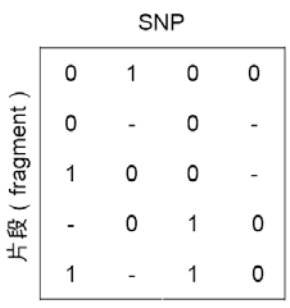


图 2.5 不可行的 SNP 矩阵

么通过 DNA 测序片段数据很容易得出个体的一对单体型。如图 2.4 所示的 SNP 矩阵，该矩阵的所有行可以划分为{行 1，行 2}和{行 3，行 4，行 5}两个子集， 显然这样的划分能使同一个子集中的任意两行在所有的 SNP 位点上均不冲突，即互相兼容，而且很容易可以得出第一个子集中的片段可以确定一个单体型 “0 1 0 0”，第二个子集中的片段可以确定另一个单体型 “1 0 1 0”。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。图 2.4 和图 2.5 中的 SNP 矩阵只有第 3 行第 3 列的 SNP 值不同，可是图 2.5 所示的 SNP 矩阵是不可行的。

基于不同的优化准则，单体型组装问题有多个不同的计算模型。

Lancia 等[9]在 2001 年的欧洲算法会议（ESA）上最先提出单体型组装问题，并引入了下面 3 个计算模型：

(1) 最少片段删除 (Minimum Fragment Removal, MFR): 给定一个 SNP 矩阵 **M**，删除最少的行使 **M** 可行；

(2) 最少 SNP 位点删除 (Minimum SNP Removal, MSR)：给定一个 SNP 矩阵 **M**，删除最少的列使 **M** 可行；

(3) 最长单体型重建 (Longest Haplotype Reconstruction, LHR)：给定一个 SNP 矩阵 **M**，删除最少的行使 **M** 可行，且使获得的两个单体型的长度和最长。

对于无空隙的 SNP 矩阵，在片段互不包含的情况下，Lancia 等[9]把 SNP 矩阵转化为有向图，然后把 MFR 转化为在该有向图中寻找两条没有公共顶点的最长有向路径问题，从而证明 MFR 是多项式可解的；随后 Lancia 等[9]通过相同的方法证明了对于无空隙的 SNP 矩阵，在片段互不包含的情况下，LHR 是多项式可解的。接着 Lancia 等[9]把无空隙的 SNP 矩阵 **M** 转化为 SNP 位点冲突图 G_S，证明了 **M** 可行当且仅当 G_S 是一个独立集（只有顶点没有边），从而把无空隙的 MSR 转化为最大独立集问题，

然后证明 G_s 是一个完美图（Perfect Graph），最后基于在完美图中寻找最大独立集是多项式时间可解的事实证明了无空隙的 MSR 是有多项式时间算法的。

对于有空隙的 SNP 矩阵 **M**, 如果 **M** 满足连续 1 属性（Consecutive ones property, C1P），因为满足连续 1 属性的 SNP 矩阵可以在多项式时间内通过重排相关的列的次序调整为无空隙的 SNP 矩阵[82,83]，因此对应的 MFR、MSR 和 LHR 也是多项式时间可解的。

对于一般的 SNP 矩阵 **M**，Lancia 等[9]把 **M** 转化成一个片段冲突图 G_F，**M** 可行当且仅当 G_F 是一个二部图。这样 MFR 就等价于删除 G_F 中最少的顶点使其二部化，而该问题是 NP-难的。然后进一步通过多项式时间归约于另外一个 NP-难 Max2SAT 问题证明了当 M 中的片段至多有一个空隙时，MFR 是 NP-难的。最后 Lancia 等[9]通过归约于 MAXCUT 问题可以证明当 M 中的片段至多有 2 个空隙时，MSR 是 NP-难的。

Rizzi 等[63]对 MSR 和 MFR 进一步进行深入研究，在 2002 年生物信息学算法会议（WABI）上提出了一些动态规划算法。对于 m 个 DNA 片段，n 个 SNP 位点的无空隙的 SNP 矩阵的 MSR 和 MFR 模型，Rizzi 等[58]提出了时间复杂度分别为 $O(mn^2)$ 和 $O(m^2n+m^3)$的多项式算法。对于有空隙的 SNP 矩阵，通过归约于图的最少边去除二部化（MinEdgeBipartizer）和最少顶点去除二部化（MinNodeBipartizer）这两个 APX-难问题，Rizzi 等[58]证明了 MFR 和 MSR 都是 APX-难。对于片段中洞的最大个数不超过 k 的 SNP 矩阵的 MSR 和 MFR 模型，Rizzi 等[58]提出了时间复杂度分别为 $O(mn^{2k+2})$ 和 $O(2^{2k}m^2n+2^{3k}m^3)$的算法。

2005 年，Bafna 等[62]在上述基础上，提出了类似的时间复杂度分别为 $O(mn^2)$和 $O(m^2n+m^3)$、空间复杂度分别为 $O(mn+n^2)$和 $O(mn+m^3)$的多项式算法求解无空隙的 MSR 和 MFR 模型；对于片段中洞的最大个数不超过 k 的 SNP 矩阵的 MSR 和 MFR 模型，Bafna 等[57]也提出了时间复杂度分别为 $O(mn^{2k+2})$和 $O(2^{2k}m^2n+2^{3k}m^3)$、空间复杂度分别为 $O(nm+2^kn^2)$和 $O(nm+2^{2k}m^3)$的动态规划算法。通过归约成顶点覆盖（Vertex cover）和最大割（Max cut）问题，Bafna 等[57]进一步证明了当片段中至多有一个空隙时，MSR 和 MFR 均是 APX-难的。

对于 LHR 模型，Cilibrasi 等[84, 85]证明了在有空隙的情况下，LHR 问题是 NP-hard 和 APX-hard。当 SNP 矩阵片段中没有空隙时，Cilibrasi[85]等设计了一个时间复杂度

为 $O(n^2m + n^3)$的动态规划算法。

单体型组装问题除了上述三个计算模型外，Lippert 等[81]曾提出了下面第 4 个计算模型：

最少错误更正 (Minimum Error Correction, MEC) 或叫做最少字符翻转 (Minimum Letter Flips, MLF)：给定一个 SNP 矩阵 **M**，求翻转（'0'变成'1'，或'1'变成'0'）**M** 中的最少元素使 **M** 可行。

对于 MEC 模型，对于一般的 SNP 矩阵，Lippert 等[81]指出可以通过归约于最大割问题证明 MEC 是 NP-难的。Cilibrasi 等[85]则进一步证明了，即使对于无空隙的 SNP 矩阵，MEC 模型也是 NP-难的，并且证明了当片段中即使只有一个空隙时，MEC 模型也是 APX-hard。

为求解 MEC 模型的精确解，R. Wang 等[63]曾设计过时间复杂度为 $O(2^m)$的分支限界算法。由于当片段数较多时，该精确算法缺乏实用性，因此有大批学者研究 MEC 模型的启发式算法。2004 年，Panconesi 等 [86]提出一种快速的启发式算法（Fast hare），在其论文中 MEC 模型也叫做 MER（Minimum Element Removal）。2004 年和 2005 年，R. S. Wang 等[63, 68]曾提出过两种动态聚类算法和一种遗传算法。2008 年，Qian 等[87]提出一种粒子群算法。

在MEC（MLF）模型的基础上，Greenberg等[88]提出了下面第5个计算模型：

WMLF (weighted minimum letter flips) 模型：给定一个SNP矩阵**M**和一个对应的权值矩阵W，翻转**M**中的元素值（0变成1，或1变成0）使得**M**可行，且W中与翻转元素对应的权值之和最小。

对于WMLF模型，Zhao等[64]证明了其是NP-难的，并给出了一个动态聚类算法。

由于个体的基因型比较容易测定，2005年，Wang等[63]则在MEC模型的基础上加入个体的基因型信息，引入了第6个计算模型：

MEC/GI (MEC with genotype information)模型：给定一个SNP矩阵**M**和基因型G，翻转**M**中最少的元素（0变成1，或1变成0），目标是使得翻转后的SNP矩阵能由一对构成G的单体型导出。

对于MEC/GI模型，R. Wang等[63]曾设计一个遗传算法和一个时间复杂度为$O(2^m)$的分支限界算法。

2006年，Zhang等[67]提出了一个与MEC/GI类似的模型MCIH（Minimum Conflict

Individual Haplotyping），证明了其是NP-难的，并提出了一个时间复杂度为$O(2^{2L}m)$的动态规划算法和一个神经网络算法FNN求解该模型，其中L为片段的最大长度。2007年，Y. Wang等[89]则提出了一个迭代的局部穷举搜索算法，Genovese等[90]则针对高测序误差和低覆盖度提出了一个启发式算法。

与上述确定性计算模型不同，Li等[91]曾于2003年提出了一种统计方法以解决单体型重建问题，该统计算法以DNA片段的碱基组成是独立的，与周围碱基取值和所在区域的位置无关等6个假设作为前提，通过计算联配的片段上不同单体型的概率来确定相邻位点上的SNP值，然后组合连接成较长的单体型段。

从上面可以看出，对单体型组装问题的研究绝大部分是研究确定性计算模型。单体型组装问题的上述计算模型绝大部分被证明为 NP-难和 APX-难的，缺乏多项式时间的精确算法和有近似度保证的近似算法，而具有较快运行速度的启发式算法无法确保算法的精确度，单体型重建精度上往往与具体的实验方法密切相关，因此设计实用的精确算法具有重要的现实意义。

2.3 k 参数化条件及 MEC/GI 参数化算法

目前最流行的 SNPs 探测方法是 DNA 直接测序[92, 93]，这种方法 48 小时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[4, 93]。当前 DNA 测序的主导方法是 Sanger 双脱氧链终止法[94]。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 ABI 3730[6]、MageBACE[7]等可测片段长度约为 1000 碱基，而 SNPs 的平均分布密度约为 1/1000[3, 95]，虽然 SNPs 在整个染色体上的分布很不均匀，从已有的数据[96-98]来看，一个长度 1000bp 的片段上的 SNP 位点是极其有限的，通常在 10 个以内。在实际应用中，一个片段上的 SNP 位点一般在 3 到 8 之间[67]。

基于以上事实，我们提出以下参数化条件：

定义2.4 k参数化条件：k是正整数，k参数化条件定义为片段覆盖的SNP位点数不超过k。

对一个 SNP 矩阵 **M** 而言，k 参数化条件等价为矩阵 **M** 的每一行最多覆盖 k 列。

[6] http://www.fungen.org/UseABI3700.htm
[7] http://www.ebiotrade.com/ custom/amersham/MegaBACE.htm

对于一个 $m \times n$ SNP 矩阵 **M** 而言，扫描 m 行可以获得 k，因此在本节中默认 **M** 满足 k 参数化条件。

2.3.1 预处理

由于MEC/GI模型中基因型G已经测定，下面根据基因型提供的信息对SNP矩阵进行预处理，以降低求其精确解的复杂性。

令 H_1 和 H_2 是构成基因型 G 的一对单体型。G、H_1 和 H_2 的第 j 个位点的值分别计作 $G[j]$、$H_1[j]$和 $H_2[j]$。

先调整调整 **M** 中各行的次序，使各行按其覆盖的最左边的列号即 l 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有序集，记作 $rowset(j)$。

然后对SNP矩阵M的每一列j，做：

Case 1. $G[j] = 0$：则$H_1[j]$和$H_2[j]$必为0，这样对于SNP矩阵的第j列上的所有值为1的单元必须翻转成0才能满足MEC/GI模型的要求。记下翻转的单元后，从M中删除该列，从G中删去第j个SNP位点的值。

Case 2. $G[j] = 1$：则$H_1[j]$和$H_2[j]$必为1，这样对于SNP矩阵的第j列上的所有值为0的单元必须翻转成1才能满足MEC/GI模型的要求。记下翻转的单元后，从M中删除该列, 从G中删去第j个SNP位点的值。

Case 3 $G[j] = 2$：则$H_1[j]$和$H_2[j]$必须不同，即如果$H_1[j] = 0$, 则$H_2[j] = 1$；如果$H_1[j] = 1$, 则$H_2[j] = 0$。对这种情况，不做任何处理。

经过上述去掉所有纯合位点的预处理后，G中留下的字符应全为2，如果没有一个字符留下，则单体型对H_1和H_2已经确定，这样很容易解决了MEC/GI问题；如果G中至少留下一个字符，则采用下面的参数化算法。在本节下面假定SNP矩阵**M**和G均通过了预处理，G的字符应全为2。为了叙述简便，用MEC/GI(**M,** G)表示对应输入**M**和G的MEC/GI解，即需要翻转的最少字符个数，如果G的字符应全为2，则G被省略。

2.3.2 K-MEC/GI 算法

对于$x, y \in \{0, 1, -\}$，令

$$d(x,y)=\begin{cases}1, & \text{如果} x \neq -, y \neq - \text{且} x \neq y;\\ 0, & \text{否则.}\end{cases} \tag{2.1}$$

给定一个$m \times n$ SNP矩阵**M**和一对单体型H = (H_1, H_2)，为了使**M**中第i行对应的片段和H_1或H_2兼容，该片段最少应该翻转的字符数记作$f_{\mathbf{M}}(i,\ \mathrm{H})$，显然有$f_{\mathbf{M}}(i,\ \mathrm{H})$ =

$\min_{p=1,2}(\sum_{j=l(i),..,r(i)} d(\mathrm{M}_{i,j}, H_p[j])$。

为了使**M**中所有行对应的片段和H中的一条单体型兼容，最少应该翻转的字符数为$\mathrm{D}(\mathbf{M}, \mathrm{H}) = \sum_{i=1...m} f_{\mathbf{M}}(i, \mathrm{H})$。

显然，对于一个$m \times n$ SNP矩阵**M**和长为n的基因型G，下面的公式成立：

$$\mathrm{MEC/GI}(\mathbf{M}, \mathrm{G}) = \min_{\mathrm{H}\text{是}G\text{的一个分解}} \mathrm{D}(\mathrm{M}, \mathrm{H}). \tag{2.2}$$

H为包含字符‘2’的基因型G的一个分解意味着，对H中的那对单体型H_1和H_2，它们每个位点上都应该是异构的，即对于任意一个SNP位点j（$1 \le j \le n$），都必定是下面两种情况之一：(1) $H_1[j] = '1', H_2[j] = '0'$； (2) $H_1[j] = '0', H_2[j] = '1'$。因此求解MEC/GI问题我们只需要考虑异构的单体型对，因此在下面的叙述中，单体型对H和L均认为是异构的。

根据 (2.2)式，对于经过预处理的$m \times n$ SNP矩阵**M**和只包含字符‘2’长为n的基因型G，其MEC/GI问题的解可以通过枚举所有可能的2^n对异构单体型来计算，所有时间为$O(mk2^n)$。当n较大时，该枚举算法是不可行的，为了降低时间复杂度，我们必须降低单体型对的搜索空间。

令$r_m(i) = \max_{p=1..i}(r(p))$，$len(i)$表示$r_m(i) - l(i) + 1$ (图2.6中包含这些符号的示例)。令$H_1[s..t]$和$H_2[s..t]$分别表示H_1和H_2表示从第s位到第t位的部分，H[$s..t$]表示单体型对$(H_1[s..t], H_2[s..t])$。因为**M**中的所有行经过了预处理中的排序，对大于i的任意行v和小于$l(i)$的任意列j，$\mathbf{M}_{v,j}$必定是空值‘−’，因此当我们从**M**的前i行构成的子矩阵出发考虑增加下一行$i + 1$时，我们只需考虑从第$l(i)$位到第$r_m(i)$位的单体型对搜索空间。

定义2.5 令i是**M**中的第i行，$\mathrm{L} = (L_1, L_2)$为一对长为$len(i)$的单体型，$F(i, \mathrm{L})$定义如下：

$$F(i, \mathrm{L}) = \min_{\mathrm{H}\text{是一对异构的单体型且H[s..t]=L}} (\sum_{q=1}^{i} f_{\mathrm{M}}(q, \mathrm{H})),$$

其中$s = l(i)$，$t = r_m(i)$。

对于经过预处理的$m \times n$ SNP矩阵**M**和只包含字符‘2’长为n的基因型G，下面的公式成立：

$$\mathrm{MEC/GI}(\mathbf{M}) = \min_{\mathrm{L}:\text{长为}len(n)\text{异构单体型对}} (F(m, \mathrm{L}), \tag{2.3}$$

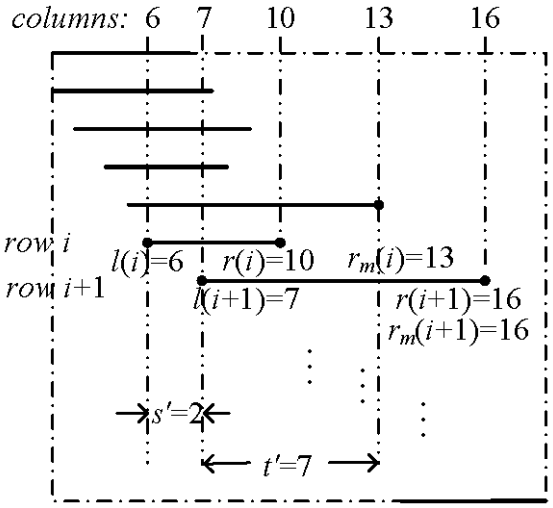


图2.6 一些记号说明

为了叙述简便，令$f'_{\mathbf{M}}(i, \mathrm{L}) = \min_{p=1,2}(\sum_{j=l(r)..r_m(i)} d(\mathrm{M}_{i,j}, L_p[j-l(i)+1]))$，容易验证

$$f_{\mathbf{M}}(i, \mathrm{H}) = f'_{\mathbf{M}}(i, \mathrm{L}), \tag{2.4}$$

其中L = H[$l(i)..r_m(i)$]。

$f'_{\mathbf{M}}(i,\ \mathrm{L})$的值可以使用函数图2.7所示的**CompF**计算，该函数运行一次所需时间为$O(len(i))$。在函数**CompF**中，异构的单体型对L用一个整数I_{L}编码，为对应L的长为$len(i)$-为的二进制数字。例如：当L = (“01001”, “10110”)时，$I_{\mathrm{L}} = (01001)_2 = 9$.

```
CompF(M, i, I_L, f', S) // I_L 是一个编码 L 的整数
{ //f' 表示 f'_M(i, L), S 表示要翻转的元素
    I' = I_L,   f'_1 = f'_2 = 0, S_1 = S_2 = Ø;
    for j = len(i) to 1 do     //从 L 中获得(L_1,L_2)
        c = I' mod 2, I'=⌊I'/2⌋;
        if c = 0 then L_1[j]='0', L_2[j]='1';
        if c = 1 then L_1[j]='1', L_2[j]='0';
    for q = 1, 2 do
        for j = l(i) to r(i) do
            if (M_{i,j} ≠ '−') and (M_{i,j} ≠ L_q[j−l(i)+1) then
                f'_q = f'_q + 1, S_q = S_q ∪ M_{i,j};
    if f'_1≤f'_2 then f' = f'_1, S = S_1;
    else f' = f'_2, S = S_2;
}
```

图 2.7　计算$f'_{\mathrm{M}}(i, \mathrm{L})$的函数 CompF

对于一对长为$len(1)$的单体型对 L, 基于定义 2.5 和(2.4)式，我们有以下等式成立：

$$F(1, \mathrm{L}) = f'_{\mathbf{M}}(1, \mathrm{L}). \tag{2.5}$$

一旦对于长为$len(i)$的任意可能异构单体型对 L，$F(i,\ \mathrm{L})$均已知，则对于一对长为

len(*i*+1)的一对单体型 L′，我们可进一步计算 F(*i*+1, L′)。令 $s = l(i+1)$, $t = r_m(i+1)$, $s' = l(i+1) - l(i) + 1$, $t' = r_m(i) - l(i+1) + 1$。这些记号示意见图 2.6。

基于定义 2.5，

$$F(i+1, \mathrm{L}') = \min_{\mathrm{H}[s..t]=\mathrm{L}'} (\sum_{q=1}^{i+1} f_{\mathrm{M}}(q, \mathrm{H})) = \min_{\mathrm{H}[s..t]=\mathrm{L}'} (\sum_{q=1}^{i} f_{\mathrm{M}}(q, \mathrm{H}) + f_{\mathrm{M}}'(i+1, \mathrm{L}'))$$

$$= \min_{\mathrm{L}[s'..len(i)]=\mathrm{L}[1..t']} (\min_{\mathrm{H}[l(i)..r_m(i)]=\mathrm{L}} \sum_{q=1}^{i} f_{\mathrm{M}}(q, \mathrm{H})) + f_{\mathrm{M}}'(i+1, \mathrm{L}').$$

因此下面等式成立：

$$F(i+1, \mathrm{L}') = \min_{\mathrm{L}[s'..len(i)]=\mathrm{L}[1..t']} (F(i, \mathrm{L})) + f_{\mathrm{M}}'(i+1, \mathrm{L}'). \tag{2.6}$$

利用(2.5)式我们可以为长为 *len*(1)的每一可能的异构单体型对 L 计算 $F(1, \mathrm{L})$。如果对于长为 *len*(*i*)的任意可能异构单体型对 L，$F(i, \mathrm{L})$均已知，则对于长为 *len*(*i*+1)的单体型对 L′，我们可利用(2.6)式计算出 F(*i*+1, L′)。对于经过预处理的 $m \times n$ SNP 矩阵 **M** 和只包含字符‘2’长为 *n* 的基因型 *G*，如果对于每个长为长为 *len*(*m*)的单体型对 L，$F(m, \mathrm{L})$已知，则根据(2.3)式我们可以得到 **M** 和 *G* 的 MEC/GI 的解。这样我们就设计出求解 MEC/GI 问题的参数化动态规划算法 K-MEC/GI。

Algorithm K-MEC/GI

Input: 一个 $m \times n$ SNP 矩阵 **M** 和一个长为 *n* 的基因型 G

Output: **M** 和 G 的 MEC/GI 问题的解

1. **预处理**: 见 2.3.1 节, 把被删除的列中需要翻转的元素记录在集合 *S* 中,令预处理后的 SNP 矩阵为 **M**′, 其行数为 m' ($m' \le m$)

//r_{m}, *len* 分别表示 $r_m(i)$和 *len*(*i*)

2. $i = 1$; $r_m = r(i)$, *len*= $r_m - l(i) + 1$;

3. **for** $I_{\mathrm{L}} = 0$ to $2^{len} - 1$ **do** //同函数 **CompF**, I_{L} 为一个编码 L 的整数

//$F[I_{\mathrm{L}}]$表示 $F(i, \mathrm{L})$, 即需翻转的元素个数，$S_{\mathrm{E}}[I_{\mathrm{L}}]$为对应的需翻转的元素集合

3.1. 调用 **CompF**(**M**′, 1, I_{L}, $F[I_{\mathrm{L}}]$, $S_{\mathrm{E}}[I_{\mathrm{L}}]$)根据(2.5)式计算 $F[I_{\mathrm{L}}]$, $S_{\mathrm{E}}[I_{\mathrm{L}}]$;

4. **while** $i < m'$ **do** //根据(2.6)式递推

//MAX 表示机器中最大的整数

4.1. $t' = r_m - l(i+1) + 1$;

//F'[L′]表示(2.6)式中的 $\min_{\mathrm{L}[s'..len(i)]=\mathrm{L}'[1..t']} (F(i, \mathrm{L}))$

4.2. **for** $I_{L'} = 0$ to $2^{t'}-1$ **do** $F'[I_{L'}] = MAX$;

4.3. **for** $I_L = 0$ to $2^{len} - 1$ **do**

4.3.1. $s' = l(i + 1) - l(i) + 1$; $L' = L[s' .. len]$;

4.3.2. **if** $F'[I_{L'}] > F[I_L]$ **then** $F'[I_{L'}] = F[I_L]$; $S'_E[I_{L'}] = S_E[I_L]$;

4.4. $i = i + 1$; $r_m = \max(r_m, r(i))$; $len = r_m - l(i) + 1$; //下一行

4.5. **for** $I_L = 0$ to $2^{len} - 1$ **do**

4.5.1. **CompF**(**M**′, i, I_L, ΔF, ΔS); $L' = L[1..t']$;

4.5.2. $F[I_L] = \Delta F + F'[I_{L'}]$; $S_E[I_L] = S'_E[I_{L'}] \cup \Delta S$; //(2.5)式

5. 在所有 $I_L = 0$ 到 $2^{len} - 1$ 中寻找使 $F[I_L]$最小的 I_L，令其为 I; 变换 **M**'的元素集 $S_E[I]$为对应的 **M** 的元素集 S_E;

6. MEC/GI(**M**,G)=$F[I] + |S|$, 对应的需翻转的元素在集合 $S_E \cup S$ 中.

定理 2.1 如果**M**满足k参数化条件,则K-MEC/GI算法能正确求解MEC/GI问题，其时间复杂度为 $O(mk2^k+mlogm+mk)$，空间复杂度为 $O(mk2^k)$。

证明：K-MEC/GI 是基于预处理和(2.3)-(2.6)式。对于经预处理后的 SNP 矩阵 **M** 及对应的基因型 G，(2.5)式的正确性可以由上面的阐述得证，其他等式的正确性也容易验证。

给定一个满足 k 参数化条件 $m \times n$ SNP 矩阵 **M**，如果对每一行 i 记录下该行覆盖的第一列和最后一列，即 $l(i)$和 $r(i)$，还有它覆盖的列上的值，那么 **M** 的存储空间为 $O(mk)$。通过第 1 步的预处理中的排序后，对于任意行 i 有 $len(i) = r_m(i) - l(i) + 1 \le k$，因此 $len \le k$；同理可知 $t' \le k$。由此可知 F 和 F'需要空间 $O(2^k)$, S_E 和 S'_E 需要空间 $O(mk2^k)$，整个算法的空间复杂度为 $O(mk2^k)$。

第 1 步预处理的时间为 $O(mk+m\log m)$。在第 3.1 步, **CompF** 需时间 $O(k)$，整个第 3 步需时间 $O(k2^k)$。第 4 步最多迭代 $m - 1$ 次需时间 $O(mk2^k)$。第 5 步需时间 $O(2^k)$。整个算法的时间复杂度是 $O(mk2^k+m\log m+mk)$。定理得证。 □

2.3.3 实验结果

对于MEC/GI模型，R. Wang等[63]曾设计一个遗传算法和一个时间复杂度为$O(2^m)$的分支限界算法B-MEC/GI和一个遗传算法G-MEC/GI。在本节我们采用同文献[63]、[79]、[86]和[99]的相同的测试方法来比较K-MEC/GI、B-MEC/GI和G-MEC/GI的性能。我们用C++语言实现了K-MEC/GI算法，Wang[63]提供了B-MEC/GI和G-MEC/GI的C++

语言程序，我们在一台Linux服务器（4个Intel Xeon 3.6G CPU，4G RAM）上对这些算法的运行时间(Running Time)和重建率(Reconstruction Rate，RR)进行了比较。

令 $\boldsymbol{h} = (h_1, h_2)$是个体一对真实的单体型，$\hat{\boldsymbol{h}} = (\hat{h}_1, \hat{h}_2)$ 是通过算法在个体的片段数据上重建出来的一对单体型，单体型重建率（Haplotype Reconstruction Rate, RR）定义[63] [64]如下：

$$\mathrm{RR}(\boldsymbol{h}, \hat{\boldsymbol{h}}) = 1 - \frac{\min(r_{1,1} + r_{2,2},\ r_{1,2} + r_{2,1})}{2n},$$

其中当i, j 等于1或2时，$r_{i,j} = \sum_{l=1,\ldots,n} d(h_i[l], \hat{h}_j[l])$，而$h_i[l]$、$\hat{h}_j[l]$分别表示这两个单体型第$l$个SNP位点上的值，$d(x, y)$的定义见(2.1)式。从上述定义可以看出，算法得出的单体型中错误的SNP数占整个两个单体型的总SNP数$2n$的比例越小，重建率RR就越高。

实验中的单体型采用 2 种方式得到，第一种与文献[63]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[8]2006 年 7 月发布的数据文件 genotypes_chr1_CEU_r21_nr_fwd_phased.gz[8]，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。第二种跟文献[86]、[92]和[99]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 d 来随机生成另一个单体型，本节采用差异率与文献[86]一样，$d = 20\%$。

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%[100]，片段的覆盖度约为 5[95, 101]。为了使模拟生成的片段数据能很好的反映真实情况，与文献[86]一样，本文采用著名的 shotgun 测序模拟数据生成器 Celsim[101]随机生成长度在 *lMin* 和 *lMax* 的片段（本节实验中 *lMin* 和 *lMax* 的默认值分别是 3 和 7），生成的片段数则按（单体型长度×片段覆盖度/片段平均长度）设置，其中片段覆盖度 c 在本节中默认为 10。最后在 Celsim 生成的片段数据的基础上，根据指定的测序误差 e 对片段的 SNP 值进行随机翻转，植入测序错误，按照空置率 p（本节中默认 p = 2%）对随机选择的 SNP 位点置空值。模拟数据生成器的详细情况请参照文献[86]和[101]。

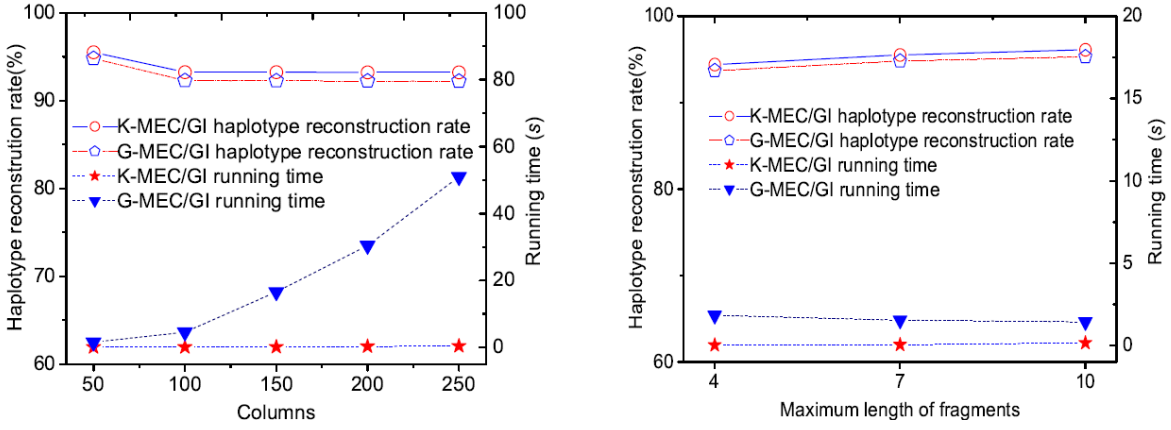
[8]从 http://www.hapmap.org/downloads/phasing/2006-07_phaseII/phased/ 下载而来

K-MEC/GI、B-MEC/GI 和 G-MEC/GI 的实验结果如表 2.1 和图 2.8 所示。

表 2.1 求解 MEC/GI 的三个算法的比较结果 [1]

Parameters			Reconstruction rate(%)			Running time (s)		
n	*m*	*e*	K-MEC/GI	G-MEC/GI	B-MEC/GI	K-MEC/GI	G-MEC/GI	B-MEC/GI
10	20	0.01	99.1(99.1)	99.1(99.1)	99.1(99.1)	0.001 (0.001)	0.18(0.19)	0.001(0.001)
		0.03	99.1(99.1)	99.1(99.1)	99.1(99.1)	0.001 (0.001)	0.18(0.19)	0.002(0.002)
		0.05	99.0(99.0)	98.7(98.6)	99.0(99.0)	0.001(0.001)	0.19(0.19)	0.011(0.004)
20	40	0.01	98.2(98.3)	97.5(97.6)	98.3(98.3)	0.008 (0.007)	0.45 (0.44)	35.12 (34.99)
		0.03	97.8 (97.5)	96.7 (96.7)	97.7 (97.4)	0.010 (0.010)	0.45 (0.44)	50.84 (51.96)
		0.05	97.2 (97.2)	96.5 (96.6)	97.3 (97.3)	0.015 (0.012)	0.47 (0.45)	83.03 (80.92)
50	100	0.01	96.9 (97.0)	95.7 (95.4)	-	0.033 (0.040)	1.48 (1.49)	>96 hours
		0.03	95.8 (96.6)	95.1 (95.2)	-	0.041 (0.042)	1.51 (1.48)	>96 hours
		0.05	95.5 (95.4)	94.8 (94.7)	-	0.047 (0.045)	1.58 (1.49)	>96 hours

[1] 括号外的是基于真实单体型的数据上的测试结果，括号里的是基于模拟单体型的数据上的测试结果，每个数据均是相同参数下 一百次重复测试结果的平均值。



(a) SNP 位点数 *n* 增加　　(b) *n* = 50，*lMax* 增加

图 2.8 K-MEC/GI 和 G-MEC/GI 性能比较

表 2.1 中括号外的数据是在真实单体型数据上的实验结果，括号内的是在模拟单体型上的实验结果。因为都是精确算法，对于一给定的 SNP 矩阵，B-MEC/GI 和 K-MEC/GI 都选择相同数量的元素翻转，但是由于所选择的元素的集合有时不唯一，因此两个算法的单体型重建率并不完全相同，当总体上没有明显的差别。当 *m*、*n* 及测序误差 *e* 度比较小时，三个算法在重构单体型时都具有很高的性能，可是当 *m* 和 *n* 增加时，G-MEC/GI 的单体型重建精度要低于 B-MEC/GI 和 K-MEC/GI。B-MEC/GI 的运行时间随 *n* 的增长而显著增长。当 *n* = 50, *m* = 100 时，K-MEC/GI 和 G-MEC/GI 的运行时间小于 2 秒，而 B-MEC/GI 却在 4 天后也没有运行完毕。

图 2.8(a)显示当 *e* = 5%，*n* 从 50 a 增大到 250, *m* 从 100 增大到 500 时，在真实单

体型数据上测试的结构。当 $n = 50, e = 5\%$, 图 2.8(b) 显示当 *lMax* 变化时真实单体型上的实验结果。在图 2.8 中，左边的 Y 坐标轴表示单体型重建精度，右边的 Y 坐标轴表示运行时间。实验结果再次表示 K-MEC/GI 比 G-MEC/GI 有更高的单体型重建精度，从图中亦可看出当 n (or m) 值较大时，K-MEC/GI 比 G-MEC/GI 要快。

2.4 单体型组装 top-*k* 枚举模型及算法

考虑到生物问题的最优解往往不是唯一的，加之由于生物现象本身的复杂性，即使在最优解只有一个的情况下，生物学家对一些接近最优的解也很感兴趣。快速的能提供最优多个解的算法，为生物学家根据领域知识做出进一步选择提供了可能，因而更能满足生物学家的需求。

2.4.1 *k*-最小距离模型

n 个 SNP 位点按在染色体上的次序从左到右记作$\{s_1, s_2, ..., s_n\}$，m 个片断记作$\{r_1, r_2, ..., r_m\}$。同 2.3.2, 两个来自于字母表{0, 1, -}的**字符 *a* 和 *b* 的距离**定义为：

$$d(a,b) = \begin{cases} 1, \text{if } a = '0' \text{ and } b = '1', \text{or } a = '1' \text{ and } b = '0'; \\ 0, \text{otherwise.} \end{cases}$$

给定一个‘0’、‘1’或‘-’组成的长为 n 的 DNA 片断 r 和一对由字母‘0’，‘1’组成的长为 n 的单体型 $H = (h_1, h_2)$，***r* 和 *H* 的距离**定义为

$d(r,H) = \min(\sum_{i=1}^{n} d(r[i], h_1[i], \sum_{i=1}^{n} d(r[i], h_2[i])$，其中 $r[i]$和 $h[i]$分别表示 DNA 片断 r 和单体型 h 的第 i 个字符。

r 和 H 的距离表示如果 r 来自于 H 代表的一对染色体，r 这个片断中最少的错误 SNP 值的个数。片段 r 和 H 的距离为 0 时称 r 和 H **兼容**。

给定一个 $m \times n$ 的 SNP 矩阵 M 和一对长为 n 的单体型 $H=(h_1, h_2)$，M 和 H 的**距离**定义为：

$dist(M,H) = \sum_{i=1}^{m} d(r_i, H)$，其中 r_i 为 M 的第 i 行。

M 和 H 的**距离**表示使得 M 中的任意片断和 H 兼容，必须修改的 SNP 值的最少个数。

基于上述定义，本文提出以下模型，使其能够为单体型组装问题提供最优的多个解。

***k*-最小距离模型(*k*-Minimum Distance)**：给定一个 $m \times n$ 的 SNP 矩阵 M 和一个正

整数 k，在所有可能的长为 n 的单体型对中，找出与 M 距离最小的 k 对单体型。

k-最小距离模型是对最少错误更正模型的扩展，一个最少错误更正模型的实例就是一个 1-最小距离模型的一个实例，由此可以证明 k-最小距离模型是 NP-难和 APX-难的。

2.4.2 算法 *k*-MD

由于 DNA 测序的错误较小，对于一个纯合的 SNP 位点，绝大部分片断在该位点的非空值应该相同，因此我们首先对 SNP 矩阵进行如下预处理：对 SNP 矩阵的每一列 j 统计取值为‘0’ 和‘1’的行数，分别记作 N_0 和 N_1。对于事先给定的一个阈值 t（本文设定为 20%），如果 $N_0/(N_0+N_1) < t$（或 $N_1/(N_0+N_1) < t$），则该列被认为是纯合的，该个体的一对单体型在该位点上取值 0（或 1）。去掉纯合的列和由此带来的空行。

经过预处理后，SNP 矩阵 M 所有的列被认为是杂合的，即所有的单体型对在经过预处理后剩下的列上取值均不相同。假设 M 的列数为 n，对应的一对杂合单体型可用一个长为 n 的在{0，1}上字符串表示。如‘0101’表示一对杂合单体型‘0101’和‘1010’。这样整个解空间为 2^n，当 n 比较大时，穷尽搜索是不可行的，因此下面采用遗传算法。

同上所述，给定一个经过预处理后的 SNP 矩阵 M，遗传算法中遗传个体采用长为 n 的在{0，1}上字符串，该字符串编码一对单体型 H。遗传个体的适应度函数为 $1-dist(M, H)/N$，其中 N 为 M 中非空值的个数。在生成下一代个体时，本文采用 Wang 等[63]的相同方法，具体算法如下。

***k*-GA 算法：**

输入：经过预处理后的 $m \times n$ SNP 矩阵 M, 要求最优的单体型对数 k；遗传算法本身参数：群体中个体个数 s, 交叉率 p_c, 变异率 p_m, 进化代数最大值 g。

输出：k 对单体型。

Step 1. 令 $l = 0$, 随机初始化群体 $P(0)=\{H_1, \ldots, H_s\}$, 即 H_i 为长 n 的在{0，1}上的随机字符串, $i = 1, \ldots, s$.

利用适应度函数计算 $P(l)$中的每个个体 H_i 的适应度，记录其中适应度最大的 k 个个体在解集 S 中。

Step 2. $l = l + 1$, 用下述方法创建 $P(l)$:

(1) 使用锦标赛选择算子从群体 $P(l-1)$中选择$(1 - p_c)\, s$ 个个体加入群体 $P(l)$

中；

(2) 使用轮盘赌选择算子从群体 $P(l-1)$中选择 $p_c\ s\ /\ 2$ 对个体随机进行单点交叉或均匀交叉操作，把由此获得的新个体加入群体 $P(l)$中；

(3) 从群体 $P(l)$中随机选择 $p_m\ s$ 个个体。对每个被选中的个体等概率进行如下操作：随机选择一个位点修改该位点的值，或者交换两个随机选择位点的值。

Step 3. 利用适应度函数计算 $P(l)$中每个个体 H_i 的适应度，如果其适应度大于 S 中个体适应度的最小值，则用 H_i 替换 S 中的具有该最小适应度的一个个体。

Step 4. 如果 $l > g$, 则算法结束，根据适应度从大到小依次返回 S 中的 k 个个体对应的 k 对杂合单体型；否则重复 Step 2 和 Step 3.

显然，把预处理中去掉的纯合列的值插入到 k-GA 算法返回的 k 对杂合单体型中，即可获得预处理前 SNP 矩阵的 k-最小距离模型的近似解。下一节称由预处理、遗传算法和后续处理组成的整个算法为 k-MD 算法。

2.4.3 实验结果

本文对 k-MD 算法求出的第一个解和来自于 Wang 等[63]的遗传算法（GA-MEC）求出的一个解进行比较测试，并对 k-MD 算法求出的多个解进行分析。k-MD 算法用 C++语言实现。实验测试在一台 Linux 服务器（4 个 Intel Xeon 3.6G CPU，4G RAM）上进行。

测试数据生成方法同文献[7, 8]，单体型采用两种方式得到：第一种采用真实的单体型数据，本文实验采用的来自于国际人类基因组单体型图计划真实单体型数据[16]，本文随机选择一个体长度 n=100 的一对单体型。第二种用计算机模拟生成，即首先随机生成长度 n=100 的单体型，然后根据 20%的差异率来随机生成另一个单体型。与文献[10]一样，在得到一对单体型的基础上本文采用著名的 shotgun 测序模拟数据生成器 Celsim[11]根据指定的参数随机生成片段数据集。在本文实验中，测序误差设置为 5%，片段数据集包含两类片断：第一类片断由 3 到 7 个连续的非空值构成；第二类片断由 7 个连续的非空值跟 10 个空值再跟 7 个连续的非空值组成。第一类片断数 m_1 由片断覆盖率 c_1 决定: $m_1=nc_1$/片断的平均长度；同样第二类片断数 m_2 由片断覆盖率 c_2 决定。模拟数据生成器的详细情况请参照文献[11]。

算法参数的设置：两个算法中群体的个体个数 s, 交叉率 p_c, 变异率 p_m, 进化代数的最大值 g 均相同，分别为 400、0.8、0.2 和 1500。对于 k-MD，k 设置为 10。

实验主要测试指标为单体型重建率 R、算法运行时间 T。单体型重建率为算法重建出的单体型对与真实单体型对具有相同值的 SNP 位点数与总的 SNP 位点数的比值。表 1 中的测试数据是相同参数下对算法重复测试 100 次结果的平均值，其中 R 为百分比，T 的单位为秒（s）。由于 k-MD 每次测试返回 10 对单体型，表 1 中 R_0 为其每次运行返回第一对单体型重建率的平均值，R_b 为其返回的 10 对单体型中最高单体型重建率的平均值。N_f 为 100 次重复测试中，k-MD 返回的第一对单体型在 10 对中具有最高单体型重建率的测试次数。

从表2.2的实验结果可以看出，当考虑其返回的第一对单体型，k-MD比GA-MEC在单体型重建精度上高出约 3%。这可能是因为这两个遗传算法的遗传个体编码方法不同，GA-MEC 的遗传个体编码空间为 2^m，而 k-MD 遗传个体编码空间最大为 2^n。当覆盖度增大时，两个算法的单体型重建精度和运行时间均有所提高。对于 k-MD 而言，返回的 10 对单体型中，第一次返回的取得最高单体型重建率的概率约为 50%，返回的 10 对单体型中最好的单体型对比第一对在单体型重建率上高出约 1%。

表 2.2 GAMEC 和 k-MDGA 性能比较

		GA-MEC		k-MD			
		R (%)	T (s)	R_0 (%)	T (s)	N_f	R_b (%)
真实的单体型数据	c_1=c_2=10	90.6	0.013	94.3	0.0041	49	95.6
	c_1=c_2=20	92.4	0.022	95.8	0.0075	46	96.8
模拟的单体型数据	c_1=c_2=10	89.8	0.0099	93.1	0.0037	54	94.2
	c_1=c_2=20	91.8	0.0208	94.6	0.0074	53	96.0

表中单体型重建率 R、R_0、R_b 和运行时间 T 为 100 次重复运行结果的平均值, c_1 和 c_2 分别是普通片段和 mate-pair 的覆盖率

2.6 本章小结

单体型组装是昂贵而耗时的直接测定单体型分子实验方法的精确而花费较少的替代方式。经过多年的深入研究，有很多单体型组装模型被提出，最少错误更正（Minimum Error Correction, MEC）是单体型的组装问题中的一个重要计算模型[102]，Wang 等[63]对 MEC 进行改进，加入个体的基因型信息，引入了 MEC/GI (MEC with Genotype Information) 计算模型。即使对于无空隙的 SNP 矩阵，MEC 模型是 NP-难，当片段数据中至少有一个空隙时，MEC 模型也是 APX-难的。在一般情况下，MEC/GI 模型也是 NP-难的。为求出上面这两个模型的精确解，Wang 等[63]曾设计了时间复杂度为 $O(2^m)$的分支限界算法，其中 m 为 SNP 矩阵的行数。对于片段数较多的情况，Wang 等[63]设计了遗传算法求其近似解。

本章根据生物实验中能直接测序的DNA片段覆盖的最大SNP位点较小的事实，对 MEC/GI 进行参数化建模，在此基础上设计出求解这个模型的精确算法K-MEC/GI。算法的时间复杂度为 $O(mk2^k+mlogm+mk)$，空间复杂度为 $O(mk2^k)$。实验结果表明， K-MEC/GI 和 Wang 等[63]的对应的分支限界算法具有相同的重构精度，而在片段数 m 达到 100，Wang 等提出的分支限界算法已无法运行的情况下，K-MEC/GI 和 Wang 等提出的遗传算法一样，仍然能快速运行。作为精确算法，K-_MEC/GI 在单体型重构精度上比 Wang 等[63]对应的遗传算法有明显优势。

考虑到生物问题的最优解往往不是唯一的，快速的能提供最优多个解的算法，为生物学家根据领域知识做出进一步选择提供了可能，更能满足生物学家的需求，因而本章设计了一个提供 top-k 个 MEC 模型优化解的遗传算法 k-MD，大量实验表明 k-MD 在单体型重建率和运行速度上要明显好于 Wang 等[63]对应的求解 MEC 问题的遗传算法 GA-MEC。

3 HLA 推断算法

3.1 前言

人类白细胞抗原(Human Leukocyte Antigen, HLA) 超级基因座(super locus)位于染色体 6p21 区域，长为 4M 碱基对，具有很高的基因密度和变异水平。0.5% (> 150 个)蛋白质编码基因位于 HLA 超级基因座内[11]，每个基因都有 10 多个不同的变异[12]。HLA 基因在免疫系统里有重要作用，它们编码称为 HLA 复合体（complex）的一组相关蛋白质，高度多态性的 HLA 基因编码出高度可变的 HLA 复合体，人类的免疫系统依靠 HLA 复合体来区分自身细胞和外界入侵细胞。器官移植中供体和受体的 HLA 基因如果不匹配引起的排斥反应会导致移植的失败。具有高度的多态性 HLA 超级基因座一直是人类遗传领域的研究热点[13]，最近很多研究者揭示 HLA 基因的不同变异与许多免疫疾病、炎症和感染有关联[14, 15]，可是用血清学和 PCR 等生物实验测试方法直接确定 HLA 基因变异耗时耗力，制约了有关 HLA 基因的大规模研究[16]，因此特别需要用有效的计算技术帮助确定 HLA 基因的不同变异。

如图3.1所示，整个HLA基因座分为class I和class II 2个传统区域加一个叫class III的中间区域。经典基因HLA-A、HLA-B和HLA-C在Class I内;经典基因 HLA-DP、HLA-DQ 和HLA-DR 在 Class II内[[13]。HLA等位基因的命名规范如图3.2所示，它们以HLA-开头后跟基因的名字和4组用冒号分开的数字及一个可选的字符后缀如‘L’、‘S’、‘C’、‘A’或‘Q’。第一组数字描述等位基因所在的组，这可以通过血清学实验确定；第二组数字描述代表编码不同蛋白质氨基酸序列的子类；最后两组数字分别表示在外显子和内含子中同义变异。最后的字符用来表示基因表达水平或其他非基

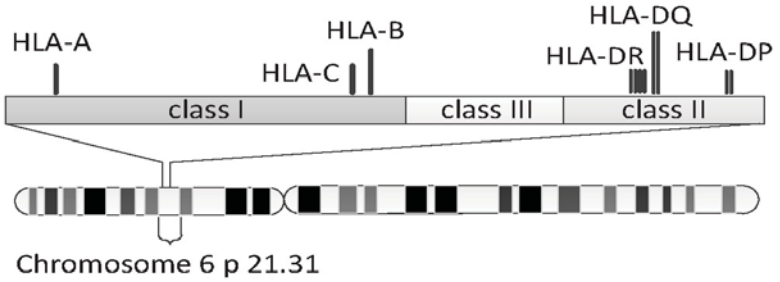


图 3.1 染色体上 HLA 区域

因组信息。虽然完整地描述一个等位基因需要4组数字，但是实际应用中通常只需要第一组或前两组数字。

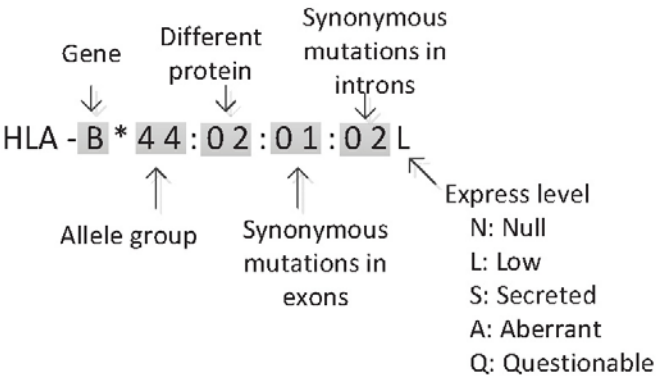


图 3.2 HLA 命名规范

随着高通量的SNP基因型分型技术的发展，全基因组SNP基因型数据可以比较容易花费较低费用测试获得，且目前已经有很多个体的全基因组SNP基因型数据可供科研工作者免费使用。最近有一些学者开始研究从SNP的基因型数据推断HLA等位基因问题。第一类方法基于标签（tag）SNP的概念，基于有多个不同取值的HLA等位基因和相邻二值SNP的连锁不平衡，de Bakker等[14]选择最多3个标签SNP作为HLA等位基因的预测因子。基于标签SNP的方法能推断出一些常见的HLA等位基因[103-105]，可它们通常为相同HLA基因的不同变异选择的标签SNP集合不同，而HLA基因是高度变异且大多数变异是不常见的，它们通常无法用三个标签SNP的不同组合来区分[16]。

通过扩展基于标签SNP的方法，Leslie等[16]提出了选择HLA基因周围几十个SNP推断类型I和II的HLA等位基因的一个统计方法，该方法认为某个HLA等位基因所在的染色体是带有该HLA等位基因的一些染色体的杂合体（mosaic）。一个隐式马尔科夫模型经过一组SNP单体型和对应的HLA等位基因值训练后用来计算一个单体型携带某个HLA等位基因的后验概率[12]，该模型需要HLA区域精细的遗传图谱[16]。

基于在一对个体中起源于同一祖先的相同序列信息（identity by descent, IBD），Setty等[12]提出了一个迭代的HLA等位基因推断算法：首先程序GERMLINE [106]被用来每一对个体之间的IBD片段信息，然后一个IBD图被创建表示一个表示人群（多数人HLA等位基因已知，少数人未知）中成对人之间IBD关系，由IBD图通过相邻已知HLA等位基因推断出未知的HLA等位基因值，最后IBD图被更新，开始新一轮的迭代，这个过程一直进行到没有更多未知的HLA基因值能被推断。虽然基于IBD图的HLA等位

基因推断方法不需要输入单体型数据，可是GERMLINE[106]需要单体型信息去计算一对个体的IBD状态。

所有上述算法HLA基因推断的准确度依赖于人群中每个个体的单体型信息的准确程度，而单体型通常是由个体的基因型数据通过某些计算模型推断出来，虽然由大量不相关的个体的基因型数据推断出每个个体的单体型的计算问题进来被广泛研究[107]，单体型特别是比较大的染色体区域(>100kb)的单体型推断的精度还不能令人满意[108]。附加的信息如家族关系能很大程度地提高较长染色体区域单体型推断的精度[109, 110]，可即使给定一个家族所有个体的基因型数据，满足孟德尔遗传规律和最少重构的个体单体型配置仍然有很多可能[109, 110]。

本章我们提出一个利用个体之间的亲缘关系、一些个体已知的HLA等位基因值和单体型和HLA等位基因之间的关系进行单体型推断进而进行HLA基因推断的统一框架。我们先提出一个新的单体型相似度量，然后在此基础上建造一个加权单体型相似图，用已知的HLA等位基因导出连接两个单体型的边的附加限制。基于不同的HLA等位基因应该有不同的单体型背景，我们提出一个算法按照一个优化原则给单体型贴上HLA等位基因的标签。为了获得个体的单体型，我们采用文献[110]提出的算法从一个家族的基因型数据按照最小重构原则构建个体单体型的解空间，为了处理家族个体不明确的单体型配置，采用一个枚举过程寻找一个最大化同一个HLA等位基因的单体型之间的相似性，如果解空间较大，枚举过程不能有效工作，我们则采用遗传算法。

3.2 基本知识

一个 SNP 认为是二值的, 通常用 0 或 1 表示, 其中 0 表示该 SNP 位点上的取值是群体中多数所取的值；1 表示群体中少数所取的值，另有一个符号‘–’表示未知值。符号 $s_1, \ldots, s_n$ 用来表示 n 个 SNP 位点，一对染色体在同一个 SNP 位点上的一对无序值(s_{i_1}, s_{i_2})表示该位点上的基因型，一条染色体上 n 个 SNP 序列叫做一个单体型记作 $h = (s_1, \ldots, s_n)$。

最近大量的实验结果显示位于HLA区域的二值的SNP和多值的HLA基因之间有很强的连锁不平衡[14]，这隐含相同的单体型可以推出相同的HLA基因取值。利用生物实验直接测定单体型跟直接测定HLA基因一样需要消耗太多的时间和金钱，而大规模SNP基因型测定要便宜得多，这样从HLA区域内SNP的基因型信息推断出HLA等位

基因是获知HLA等位基因的另一种有吸引力的方法。

指定一个HLA基因，给定一个群体在这个基因附近SNP的基因型数据和该群体中的一些个体该基因的取值，**HLA基因推断（HLA gene type inference）问题**是要确定其余个体该基因的取值，图3.3就是一个HLA基因推断问题的图示。HLA基因推断问题的输入数据中n个个体在m个SNP位点上的基因型数据用一个$n\times m$ 基因型矩阵表示，矩阵中的每一个元素均是一对无序的SNP值；人群的HLA基因取值信息用一个$n\times 2$的HLA矩阵来表示，其中未知的HLA基因取值用‘–’。在图3.3中第2个和最后一个HLA-A的基因取值信息在输入是未知的（图的中间），它们的值通过HLA基因推断算法获得（图的右边）。

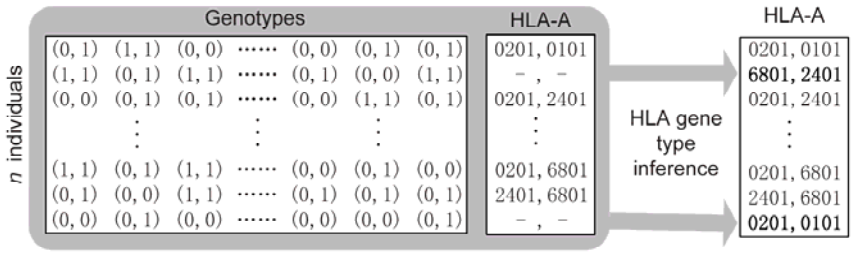


图3.3 HLA基因推断（HLA gene type inference）问题示例

尽管从基因型推断单体型的算法有很多，这些算法对无关个体组成的群体进行的单体型推断精度很不令人满意，但是当已知一个家族所有成员的基因型信息时，单体型推断的精度会明显改善[107]，因此Leslie等[16]和Setty等[12]在测试他们的算法时就采用家族数据。然而即使对于一个家族，唯一地为每个成员确定单体型配置也十分困难[110]，在文献[110]中，作者设计了一个程序DSS为一个家族根据孟德尔遗传规律和零重构限制确定个体单体型配置的解空间，DSS使用分离集D（disjoint-set）表示整个解空间，解空间内解的个数N_s取决定于自由变量的个数f：$N_s = 2^f$。为了处理重构，DSS进行了下面的扩展：首先整个区域分为几个极大零重构子区域，然后这些子区域单体型解空间联结起来形成一个完整的解空间。由于DSS快速且精度较高，在下述算法中我们采用DSS进行单体型推断。

3.3 WSG-HI 算法

给定一些家族的基因型数据和其中多数个体的 HLA 基因数据，算法 WSG-HI（HLA gene type Inference based on a weighted Similarity Graph） 为 HLA 推断问题建

模和单体型优化选择建立一个统一的框架。我们首先定义一个新的单体型相似度, 然后构建一个顶点表示单体型的完全加权图，图中的连接着一对单体型边的权重就是这对单体型的相似度，除此之外，对每一个 HLA 基因已知的个体（其 HLA 基因对已知），对应的两个单体型之间增加一条限制边，边上标上对应的 HLA 基因对，表示这两个单体型的 HLA 基因赋值混合起来应该等于该限制边上的标签。构建单体型相似图的目标是对每个顶点（单体型）进行 HLA 基因的优化赋值，使得该赋值满足限制边的约束，又能使具有同样 HLA 基因赋值的顶点之间的边权重之和最大。当一个家族有多个不同的单体型配置时，我们将选择其中一个使相同 HLA 基因赋值的顶点之间边权重和最大的配置。

具体来说，算法 WSG-HI 主要包含两步：第一步搜索 HLA 基因和背景单体型之间的优化关系，同时从每个家族多个可能的单体型配置中选择一个最优的配置；第二步对那些HLA基因未知的个体相关的单体型进行HLA基因优化赋值。为了叙述简单，HLA 基因和背景单体型之间的关系称为 **Hap-HLA 关系**。在第一步中，扩展的 DSS[110] 用来确定一群人 P 中每一个家族的单体型配置解空间。在很多情况下，P 的整个解空间太大而无法直接枚举，因此 WSG-HI 采用增量方式处理整个人群 P，我们首先选择只有唯一的单体型配置的家族进行处理，得到对应的单体型相似图，计算一个部分 Hap-HLA 关系。其余的家族按单体型配置解空间大小从小到大依次逐个进行处理，当一个家族的单体型配置解空间较小时，这个家族的每个可能的单体型配置被枚举并组合到已经获得的单体型相似图中，然后在计算出一个度量值，具有最高值的解被选择为该家族的单体型配置；当解空间太大而无法枚举时，遗传算法被采用为该家族选择一个最优的单体型配置。上述过程迭代进行直至所有家族处理完毕。最后在第二步利用在第一步确定的Hap-HLA关系根据单体型相似图为HLA基因未知的单体型确定其对应的 HLA 基因取值。算法细节将在下面详细描述。

3.3.1 单体型相似性

令 $h_i = (s_{i1}, s_{i2}, \ldots, s_{in})$和 $h_j = (s_{j1}, s_{j2}, \ldots, s_{jn})$ 分别为长为 n 的两条单体型, 其中 s 是一个 SNP。如果 s_{il}，$s_{jl} \neq$ ‘−’ 且 $s_{il} \neq s_{jl}$，那么在位点 l 单体型 h_i 和 h_j 不匹配；如果 s_{il}，$s_{jl} \neq$ ‘−’ 且 $s_{il} = s_{jl}$，那么在位点 l 单体型 h_i 和 h_j 匹配。给定一个阈值 T_{mis}, 如果下面条件满足，$[p, q]$定义为 h_i 和 h_j 的一个极大匹配区间：

(1) $1 \leq p < q \leq n$;

(2) h_i 和 h_j 在区间的两个端点匹配;

(3) 在该区间 h_i 和 h_j 的连续不匹配位点数不超过 T_{mis};

(4) 不存在一个包含[p, q]的更大区间[p', q']满足条件(2)和(3)。

当允许基因型数据中存在错误时，T_{mis} 的取值应大于 0。令 h_i 和 h_j 的所有极大匹配区域的集合为 $S_r(h_i, h_j)$, h_i 和 h_j 在[p, q]内匹配的位点数为 $N_{pq}(h_i, h_j)$，h_i 和 h_j 的**相似度**定义为：

$$Similarity(h_i, h_j) = \frac{\max_{[p,q]\in S_r(h_i,h_j)} N_{pq}(h_i, h_j)}{n}. \tag{3.1}$$

从直觉上说，$Similarity(h_i, h_j)$反映了单体型 h_i 和 h_j 与相同 HLA 基因相关联的似然度。当 h_i 和 h_j 完全相同, $Similarity(h_i, h_j)$取最大值 1，这表示有很高的概率 h_i 和 h_j 具有相同的 HLA 基因。

3.3.2 加权相似图

给定一个单体型集合H，对应的加权相似图G_H构建方法如下。对每一个个体的一对单体型(h_i, h_j)及一对HLA基因{α, β}，G_H中有两个顶点i和j及一条连接这两个顶点的**限制边**c_{ij}，c_{ij}的限制标签为{α, β}，如果$\alpha \neq \beta$, 则c_{ij}为一条**异构**限制边；如果$\alpha = \beta$则c_{ij}为一条**同构**限制边。任意一对顶点p和q之间有一条相似性边e_{pq}，其权值w_{pq} = $Simlarity(h_p, h_q)$。G_H的顶点集记作V (G_H) ，在所有限制边的HLA基因的集合记作$C(G_H)$。图3.4(a)是一个加权相似图示例，该图表示有两个个体I_1和I_2，他们的单体型分别是{h_1, h_2}和{h_3, h_4}，他们的HLA基因取值分别是HLA-A{0201, 0101}和{0201, 2401}。

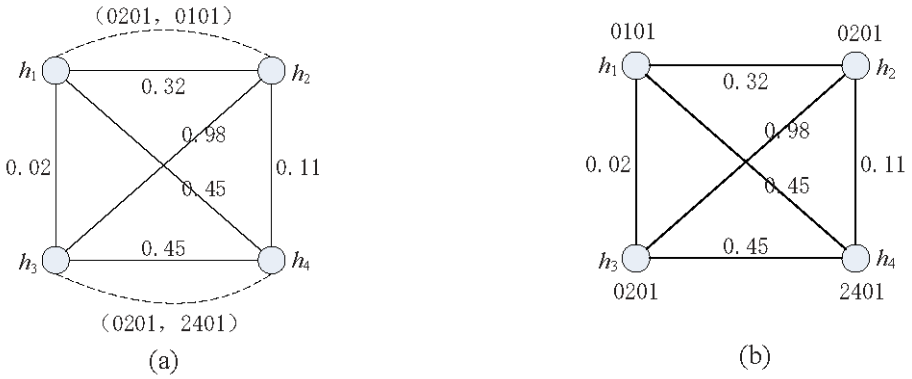


图3.4 一个加权相似图(a)及它的一个可行标签形式(b)

3.3.3 加权相似图优化标签

加权相似图G_H的一个标签函数l: $l(i)=\alpha$ 是从$V(G_H)$到$C(G_H)$的一个映射，其中$i \in V(G_H)$，$\alpha \in C(G_H)$, $l(i)=\alpha$表示对G_H中的顶点i（或对应的单体型h_i）赋予HLA 基因型α。对于一条限制标签为$\{\alpha, \beta\}$的限制边c_{ij}，其关联的两个顶点为i和j，如果$l(i) \cup l(j)=\{\alpha, \beta\}$，则限制边$c_{ij}$得到满足。如果标签函数使所有的限制边得到满足，则$l$是**可行**的并给出了一种Hap-HLA关系。按照一个可行的标签函数l给G_H的顶点贴上标签并去掉所有的限制边我们就得到了G_H的一个可行标签形式G_H^l（图3.4(b)是图3.4(a)的一个可行标签形式）。图G_H可以有很多不同的可行标签形式，不同的标签形式代表不同的Hap-HLA关系，为了选择其中最优的一个，我们定义如下的优化目标函数：

$$Con(G_H^l)=\sum(w_{pq} \mid p,q \in V(G_H^l) \wedge l(p)=l(q)). \tag{3.2}$$

加权相似图G_H的优化标签问题是发现一可行的标签函数l使$Con(G_H^l)$最大。当G_H有N_h条异构限制边，简单的强力搜索算法需要时间$O(2^{N_h})$求解G_H的优化标签问题，当N_h较大时，强力搜索算法显然是不可行的。为了在较短的时间内求解该问题，我们设计了图3.5所示的启发式算法**Heu-Label**。在该算法的Step 2中，与所有同构限制边相关联的顶点都被无二义地标上对应的HLA基因，所有的同构限制边均被删除。在Step 3，一个指定的阈值T_s用来删除权值较小的边期望能得到一个稀疏图，在我们的实验中，当T_s从0.55变化到0.90时，WSG-HI算法性能仅发生较小的变化，在进行实验测试

Input: A weighted similarity graph G_H.
Output: A labeling l of G_H.
Step 1: **for** each vertex i of G_H **do** { $l(i)$ = '–'; }
Step 2: **for** each homozygous constraint edge c_{ij} of G_H **do**
α = an HLA gene type in the constraint of c_{ij}; $l(i)=l(j)=\alpha$; delete c_{ij} from G_H;
Step 3: build a graph G by deleting all constraint edges and the similarity edges whose weights are small than T_s from G_H;
Step 4: find all connected components of G by depth first search;
Step 5: **for** each connected component *comp* (from the largest to the smallest) **do**
Step 5.1: **for** each HLA gene type α in $C(G_H)$ **do** { $N(\alpha)=0$;}
Step 5.2: **for** each vertex i in *comp* **do**
if $l(i) \neq$ '–' **then** $N(l(i))++$;
else { (α,β) = the constraint of the constraint edge adjacent to i in G_H; $N(\alpha)$++; $N(\beta)$++;}
Step 5.3: $\gamma=\underset{\alpha}{\text{argmax}}(N(\alpha))$;
Step 5.4: **for** each vertex i in *comp* **do**
(α,β) = the constraint of the constraint edge c_{ij} adjacent to i in G_H;
if $l(i)$ = '–' **then**
if $\alpha=\gamma$ **then** $l(i)=\alpha$; $l(j)=\beta$; delete the constraint edge c_{ij} from G_H;
delete vertices i and j from G;
if $\beta=\gamma$ **then** $l(i)=\beta$; $l(j)=\alpha$; delete the constraint edge c_{ij} from G_H;
delete vertices i and j from G;
Step 6: repeat Steps 4 and 5 until there are no more vertices can be labeled.

图3.5 Heu-Label的伪代码

时， T_s默认设定为0.65。在后续的步骤中，顶点标签按照在同一个联通分量大部分顶点用同样HLA基因标签的启发式原则进行，细节请参见图3.5。为了简便，由Heu-Label获得的G_H的一个可行标签形式记为$G(H)$。

3.3.4 优化单体型配置和 Hap-HLA 关系

在本小节中，我们讨论逐步增加有多个单体型配置的家族的细节。假设对于一个子人口P'一个优化的单体型配置H'和一个Hap-HLA关系l'已经得到，这些信息都包含在图$G_{H'}^{l'}$中。当一个新的家族P''要加入到P'时，搜索$P' \cup P''$的优化单体型配置H和Hap-HLA关系l的方法如下。

使用扩展的DSS程序于新来的家族P''，获得描述P''单体型配置解空间的一个不相交集（disjoint-set）D。通过给一个2值自由变量向量$S = (v_1, \ldots, v_f)$赋值，基于D我们可以获得P''一个单体型配置，其中f是解空间的自由变量个数。当P''的不同单体型配置总数不大于2^{Tc} (在我们的实验中$T_c = 10$)，一个图3.6所示的简单穷尽枚举过程**Enum-Alg**用来挑选其中的一个单体型配置。对每一个P''的每一个不同单体型配置，Enum-Alg添加对应的顶点和边到$G_{H'}^{l'}$进而获得了一个更新的图$G_{H'\cup H''}^{l'}$，通过调用Heu-Label，我们可以获得该图的一个新的标签函数l。在所有的单体型配置中，Enum-Alg选择一个配置具有最大化的$Con(G_{H'\cup H''}^{l})$。

```
Enum-Alg(G^{l'}_{H'}, P'')
{   for each haplotype configuration H'' of P'' do
    {   construct a graph G^{l'}_{H'∪H''} by adding two vertices for each individual of P'', the corresponding
similarity edges and constraint edges in G^{l'}_{H'};
        apply Heu-Label to G^{l'}_{H'∪H''} to obtain a labeling l for G^{l'}_{H'∪H''};
    }
    return the G^{l}_{H'∪H''} with the maximum Con(G^{l}_{H'∪H''});
}
```

图3.6 Enum-Alg的伪代码

当P''的不同单体型配置总数太大（即$f > T_c$）时，我们采用图3.7所示的**Genetic-Alg**遗传算法。Genetic-Alg直接使用DSS程序使用的解空间自由变量向量S作为遗传算法中个体编码，该编码唯一确定P''的一个单体型配置，记作$H(S)$。Genetic-Alg采用的适应性函数（fitness function）是$Con(G(H' \cup H(S)))$。遗传编码空间为$\{(v_1, \ldots, v_f) \mid v_i \in \{0, 1\}, i = 1, , \ldots, f\}$，遗传选择算子采用锦标赛和轮盘赌[111]，单点变异和单点交叉用来产生新的遗传个体，在我们的试验中，Genetic-Alg的参数设置如下：遗传个体数p_s

= 400, 遗传进化最大代数$g_m = 50$，交叉率$r_c = 0.8$，变异率$r_m = 0.2$。

```
Genetic-Alg(G_{H'}^{I}, P'')
{   //p_s: population size, r_c: crossover rate, r_m: mutation rate.
    //g_m: the maximum number of population generation.
    //f: the number of free variables to describe the haplotype configuration space of P''.
    randomly generate p_s individuals (i.e., f dimensions vector of {0,1}) to form a population P_0;
    for i = 1 to g_m do
    {   for each individual S ∈ P_0 do
        {   compute the fitness of S, i.e., Con(G(H' ∪ H(S)));
            if the fitness of S is maximum so far then S_m = S ;
        }
        if the fitness of S_m remains unchanged or i = g_m then return G(H' ∪ H(S_m));
        // produce a new generation of population
        P_1 = ∅;
        select (1 − r_c) × p_s individuals from P_0 into P_1 using the tournament selection operator;
        select r_c × p_s/2 pairs of individuals from P_0 using the roulette wheel selection operator into M;
        for each pair (i, j) ∈ M do
        {   randomly apply single-point crossover operator on (i, j) and put the two offspring into P_1;   }
        select r_m × p_s members from P_1 randomly and invert the value at a random position for each selected
member;
        P_0 = P_1;
    }
}
```

图3.7 Genetic-Alg的伪代码

3.3.5 HLA 基因推断

令R为HLA基因已知的个体的集合，U是其他个体的集合，$H(I)$表示个体I的单体型对, $H_1(I)$和$H_2(I)$为$H(I)$中的两条单体型, $V_1(I)$和$V_2(I)$是G_H中对应$H_1(I)$和$H_2(I)$ 的两个顶点，$w_m(i)$表示邻接顶点i的边的最大权值。对于一对HLA基因$\{g_1, g_2\}$，令$w_m(I; g_1, g_2)$表示

$$\max_{I' \in R \wedge l(V_1(I'))=g_1 \wedge l(V_2(I'))=g_2} (w_{V_1(I)V_1(I')} + w_{V_2(I)V_2(I')}).$$

当执行完Heu-Label之后，如何集合U非空，我们采用基于相似的单体型具有相似的HLA基因的原则对一个属于U的个体I对应的顶点进行HLA基因赋值。首先如果$w_m(V_1(I))$或$w_m(V_2(I))$比阈值T_s (在实验测试中T_s为0.65)小，个体I的HLA基因不能推断出来，否则I的HLA基因按照如下方法进行推断。

令$L(V_1(I))$和$L(V_2(I))$分别为和顶点$V_1(I)$和$V_2(I)$通过具有最大权重的相似边邻接顶点的HLA基因集合，即：

$$L(V_1(I)) = \{l(p) \mid w_{pV_1(I)} = w_m(V_1(I)) \wedge l(i) \neq \text{‘}-\text{’}\};$$

$$L(V_2(I)) = \{l(p) \mid w_{pV_2(I)} = w_m(V_2(I)) \wedge l(i) \neq \text{‘}-\text{’}\}.$$

如果$L(V_1(I))$ (或$L(V_2(I))$)只包含一个元素，则顶点$V_1(I)$ (或$V_2(I)$)就用该元素作标

签，这样就确定了单体型$H_1(I)$ (或$H_2(I)$)的HLA基因。如果在$L(V_1(I))$ (或$L(V_2(I))$)中有多个元素， 满足下面条件的HLA基因g_1和g_2就分别用作$V_1(I)$和$V_2(I)$的标签：$g_1 \in L(V_1(I))$, $g_2 \in L(V_2(I))$且$w_m(I; g_1, g_2)$为最大，具体细节请看图3.8的HLA-type子程序的伪代码。

```
HLA-type(G_H, R, U)
{   for each I ∈ U do
    {   i = V_1(I); j = V_2(I);
        w_m(i) = max_k(w_ik);
        w_m(j) = max_k(w_jk);
        L(i) = L(j) = ∅;
        if w_m(i) < T_s or w_m(j) < T_s then
        {   l(i) = '–';   l(j) = '–';   continue;   }
        for each vertex k ∈ R do
        {   if w_ik = w_m(i) then L(i) = L(i) ∪ l(k);
            if w_jk = w_m(j) then L(j) = L(j) ∪ l(k);
        }
        if |L(i)| = 1 then l(i) = the element in L(i);
        if |L(j)| = 1 then l(j) = the element in L(j);
        if |L(i)| > 1 or |L(j)| > 1 then
        {   traverse all constraint edges of G_H to compute w_m(I; g_p, g_q) for g_p ∈ L(i) and g_q ∈ L(j);
            (g_1, g_2) = argmax_{g_p∈L(i), g_q∈L(j)} w_m(I; g_p, g_q);
            l(i) = g_1; l(j) = g_2;
        }
    }
}
```

图3.8 HLA-type的伪代码

完整的WSG-HI算法的伪代码如图3.9所示，WSG-HI的运行时间主要取决于在人口P中各家族的可能的单体型配置个数。

3.4 实验结果

我们在一台1GB内存2.4GHz CPU的Linux PC机上运行WSG-HI，测试数据来自于文献[4]，可以从http://www.inammgen.org/inammgen/_les/data/公开下载。该数据来自于HapMap计划的CEPH数据集，其包含180个欧洲血统的犹他州居民，分属27个家庭，一个家庭的平均人数为6.6个，详细描述请参考文献[4]。这些个体位于扩展的HLA区域中8562个SNP上的基因型被测定，其中6300个通过质量控制[16]，类I中的三个HLA基因(HLA-A, HLA-B, HLA-C)和类II中的三个HLA基因(HLA-DRB1, HLA-DQA1, HLADQB1)通过PCR-SSOP协议测定。测试数据提供一对染色体上混合的SNP数据和HLA基因数据，单条染色体的分型数据都是未知的。

INPUT: genotype matrix M_G and HLA matrix M_H of a population P made up of pedigrees $p_1, \ldots, p_k$.

OUTPUT: inferred HLA gene types for the individuals whose HLA gene types are unknown.

STEP 1: (find out an optimal Hap-HLA relation for P)

Step 1.1: **for** $i = 1, \ldots, k$ **do** apply the extended DSS to obtain a disjoint-set structure D_i and free variables $v_1, \ldots, v_{f_i}$ that describe the solution space of the haplotype configuration of pedigree p_i;

Step 1.2: sort the pedigrees in P in ascending order according to f_i;

Step 1.3: $H' = P' = \varnothing$; $i = 1$;

Step 1.4: **while** $f_i = 0$ and $i \leq k$ **do** $\{P' = P' \cup p_i; H' = H' \cup$ the unique haplotype configuration of p_i; $i++;\}$

Step 1.5: build a weight similarity graph $G_{P'}$ for P' using M_G and M_H ;

Step 1.6: apply procedure Heu-Label to $G_{P'}$ and obtain $G(H')$ using M_G and M_H;

Step 1.7: **while** $f_i < T_c$ and $i \leq k$ **do**

$\{ G(H') =$Enum-Alg $(G(H'), p_i); i++;\}$

Step 1.8: **while** $i \leq k$ **do**

$\{ G(H') =$ Genetic-Alg$(G(H'), p_i); i++;\}$

Step 2: (infer the HLA gene types for the individuals whose HLA gene types are unknown)

Step 2.1: scan M_H to calculate the set R of the individuals in P whose HLA gene types are known and the set U of the other individuals;

Step 2.2: HLA-type$(G(H'), R, U)$;

Step 2.3: the HLA gene types of an individual $I \in U$ with the haplotype configuration (h_p, h_q) is $(l(p), l(q))$;

图3.9 WSG-HI算法的伪代码

按照文献[12]，算法的性能用覆盖率Coverage和精度Accuracy来分析和比较，它们定义如下：

$$Coverage = \frac{N_{called}}{N_{analyzed}}, \tag{3.3}$$

$$Accuracy = \frac{N_{correct}}{N_{called}}, \tag{3.4}$$

其中$N_{analyzed}$表示需要进行HLA基因推断的染色体总数，N_{called}表示被算法推断出HLA基因的染色体数，$N_{correct}$表示被算法正确推断出HLA基因的染色体数。测试时对于每个HLA基因，WSG-HI取以该基因为中心的长200kb区域的SNP基因型数据和带4位数字或2位数字的HLA基因作为输入。测试结果不考虑测试数据中没有达到指定位数的或在数据集中只出现一次的HLA基因。

3.4.1 Leave-one-out 测试结果

我们采用Setty等[12]所用的Leave-one-out测试WSG-HI的性能，然后和他们提出的算法IBD-HI进行性能比较。WSG-HI需要预先确定2个阈值T_{mis}和T_s。我们首先固定T_s = 0.65，然后T_{mis}的值从1改变到2和3来测试WSG-HI的性能，实验结果如表3.1所示。从实验结果可以看出当T_{mis}取值较小时，WSG-HI的性能比较稳定，没有显著的变化，默认情况下WSG-HI设定T_{mis}为2。当固定T_{mis} = 2，T_s从0.55增大到0.90时，我们进一步测试WSG-HI的性能，实验结果如表3.2所示。从实验结果可以看出当T_s从0.55增大到0.90时，WSG-HI的性能只发生轻微变化，默认情况下WSG-HI设定T_s为0.65。

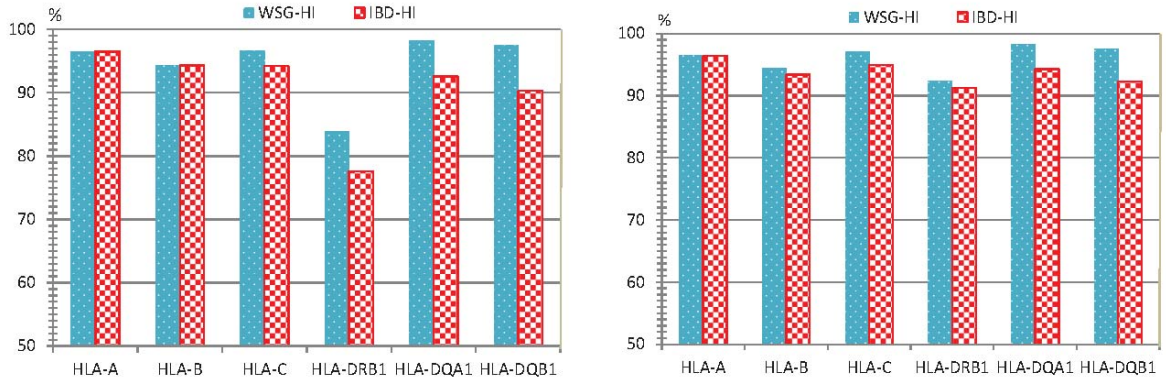
表3.1 T_s = 0.65, T_{mis}=1, 2, 3时WSG-HI推断带4位数字HLA基因的精度.

	Accuracy(%)		
T_{mis} =	1	2	3
HLA-A	95.89	96.50	96.50
HLA-B	93.57	94.31	94.26
HLA-C	94.82	96.65	96.34
HLA- DRB1	84.19	83.87	84.52
HLA- DQA1	98.00	98.29	98.29
HLA- DQB1	97.14	97.43	97.71

表3.2 T_{mis}=2, T_s 从0.55增大到0.90时WSG-HI推断带4位数字HLA基因的精度.

	Accuracy(%)							
T_s =	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90
HLA-A	95.89	95.89	96.50	95.89	95.89	95.89	95.89	95.89
HLA-B	95.00	94.64	94.31	94.64	94.64	94.64	94.64	94.64
HLA-C	96.34	96.34	96.65	95.73	96.04	95.43	95.12	93.29
HLA- DRB1	83.87	83.87	83.87	83.87	83.87	83.87	83.87	83.87
HLA- DQA1	98.29	98.29	98.29	97.71	98.00	97.71	97.71	96.57
HLA- DQB1	97.14	97.43	97.43	97.14	97.14	96.86	97.14	96.86

我们进而对阈值取默认值时WSG-HI的性能和IBD-HI的性能进行比较，比较结果如图3.10所示。在所有实验结果中WSG-HI的coverage都达到了100%而Setty等并没有提供他们算法的覆盖度数据[12], 因此我们在图3.10中仅对两个算法的精度进行比较。从实验结果可以看出，在Leave-one-out测试中，除了在推断HLA-DRB1基因时，两个算法的HLA基因推断精度都比较高，大于或等于90%，两个算法推断带两位数字的HLA基因的精度比推断带4位数字的HLA基因的精度要高。在对HLA-C、HLA-DBR1、HLA-DQA1、HLA-DBQ1的推断中，WSG-HI的精度明显高于IBD-HI。



(a) Accuracy at 4-digit resolution　　(b) Accuracy at 2-digit resolution

图3.10 算法WSG-HI和Setty等的算法IBD-HI的性能比较。(a) HLA基因名称带4位数字；(b) HLA基因名称带2位数字。IBD-HI的精度数据直接来自于文献[2].

由于WSG-HI采用了遗传算法，为了考察遗传算法中的随机因素对WSG-HI性能的影响，我们对WSG-HI进行了重复测试，其HLA基因推断精度和运行时间如表3.3所示，从表中可以看出WSG-HI具有很强的稳健性。

表3.3 5次重复测试WSG-HI推断带4位数字HLA基因的精度和运行时间.

	Accuracy (%)					*Running time* (minutes)				
	1	2	3	4	5	1	2	3	4	5
HLA-A	96.50	96.20	96.82	96.50	96.20	55.5	56.4	56.3	56.2	56.1
HLA-B	94.30	94.66	95.00	95.00	95.00	47.3	47.6	47.7	47.6	47.6
HLA-C	96.65	96.65	96.65	96.34	96.34	58.9	59.7	59.8	59.9	59.0
HLA- DRB1	83.87	84.19	84.19	83.87	83.87	41.6	41.6	41.5	41.5	41.5
HLA- DQA1	98.29	98.57	98.29	98.57	98.57	29.8	29.7	29.7	29.8	29.7
HLA- DQB1	97.43	97.71	97.71	97.43	98.00	37.3	38.1	38.1	38.1	38.1

3.4.2 Leave-one-pedigree-out 测试结果

因为测试数据包含许多家族（pedigree），在leav-one-out测试中只有一个人的HLA基因不知，从直觉上感觉算法能比较容易地从该个体所在家族的其他成员HLA基因正确推断出未知的HLA基因。为了进一步评估WSG-HI的性能，我们继续对算法进行Leave-one-pedigree-out测试，即每次测试时，随机选择一个家族，隐藏其所有个体的HLA基因信息，然后测试WSG-HI推断该家族所有个体HLA基因的精度，实验结果如表3.4所示。令人吃惊的时，WSG-HI在Leave-one-pedigree-out测试中性能同在Leave-one- out测试中性能几乎一样好，这可能是因为测试数据集中的其他家族提供了足够的单体型相似信息给WSG-HI进行HLA基因推断，虽然在Leave-one- out测试提供了同家族其他成员的信息，但是WSG-HI并没有利用这种家族关系进行HLA基因推断。

表3.3 在Leave-one-pedigree-out测试中WSG-HI推断带4位数字HLA基因的精度

	4-digit		2-digit	
Gene	*Coverage*(%)	*Accuracy*(%)	*Coverage*(%)	*Accuracy*(%)
HLA-A	100	95.6	100	95.3
HLA-B	100	93.2	100	93.0
HLA-C	100	95.6	100	95.3
HLA-DRB1	100	80.8	100	92.4
HLA-DQA1	100	93.5	100	94.9
HLA-DQB1	100	94.0	100	94.2

3.5 本章小结

HLA 基因在人类免疫系统有重要的作用，它们的变异和许多复杂疾病相关联，可是直接用生物学实验确定 HLA 基因在时间和金钱上代价过高，因此如果能从 SNP 基因型数据精确而高效地推断出 HLA 基因将会进一步推动 HLA 基因的相关研究。在本章中，我们设计了一个 HLA 基因推断算法 WSG-HI，该算法根据一个群体的 SNP 基因型数据和其中多数个体的 HLA 基因信息推断出其余个体的 HLA 基因。在一组预先测定了 SNP 基因型和 HLA 基因的数据集上的大量实验测试显示 WSG-HI 能根据 HLA 基因相邻区域的 SNP 基因型数据精确地进行 HLA 基因推断。和最近提出的一个基于 IBD 的算法[2]相比，WSG-HI 在推断 HLA-A 和 HLA-B 时达到了相同的精度，在推断 HLA-C、HLA-DRB1、HLA-DQA1 和 HLA-DQB 时，WSG-HI 更精确。

4　基于相对频繁项聚类探测多基因交互

4.1 前言

2002 年一个基因芯片能同时为 1 万个 SNP 位点进行基因分型，而 2007 年一个基因芯片同时进行基因分型的 SNP 位点数目增长了 100 倍，为 1 百万个 (Altshuler et al., 2008)。随着基因分型技术的飞速发展，在最近五年内科研机构收集的常见疾病表型信息及相关个体的全基因组基因型信息呈加速度增长[17-19]。利用大量个体全基因组 SNP 位点上的基因型信息和相关的疾病表型信息进行全基因组关联分析 (Genome-wide association studies, GWAS)是揭示复杂疾病致病基因的有效手段[20, 21]。目前 GWAS 采用的主要模式是疾病与单个 SNP 位点相关统计分析的方法[22]，可是人类复杂疾病往往是多个基因的交互作用(epistasis)的结果[23, 24]。大量研究结果显示乳腺癌[25]、糖尿病[23]和冠心病[26]等人类常见疾病与多个基因的交互作用有密切关系。基于单个 SNP 位点的统计方法可能无法探测到所有交互的基因，特别是在交互的多个基因中单个基因变异不能显著影响疾病的发病率的情况下。

最近，众多科研工作者对探测全基因组 epistasis 问题进行了大量研究，提出了多个算法[23, 112-118]. 这些搜寻 SNP-SNP（或基因-基因）交互作用的方法可分为以下四类：穷尽搜索，随机算法，数据挖掘与机器学习，步进搜索。基于穷尽搜索的 epistasis 算法枚举所有可能的多个 SNP 组合并进行相关的与疾病的交互测试（如卡方测试或对数回归）。Nelson 等[26]提出的组合分区方法 CPM 搜索所有可能的把 m 个基因型组合分成 k 组的方式，从中找出能解释表型数量症状值的最佳划分方式。由于巨大的划分空间，CPM 只适合很小数据集的两位点交互探测。受 CPM 的启发，Ritchie 等[25]提出了多维约简算法（multifactor-dimensionality reduction, MDR)。 MDR 把多位点基因型组合空间划分为两类，穷尽搜索所有可能的划分，寻找出能最准确预测疾病状态的多位点基因型划分模型。MDR 使用重复的交叉验证（cross-validation）和交换测试（permutation test）来评估基因型划分模型的预测精确度和统计显著性， 而交叉验证（cross-validation）和交换测试（permutation test）需要大量的计算资源，因此

同 CPM 一样，即使只考虑两位点交互，MDR 也不能处理较大的数据集(23)。最近有许多研究者对 MDR 进行扩展，提出了如 MB-MDR[119] 和 RMDR[120]等算法，可是它们仍然不能有效处理全基因组关联分析数据。为了提高算法效率，2010 年 Wan 等[117]提出了基于布尔操作的筛选和测试方法（boolean operation-based screening and testing，BOOST)，BOOST 能有效地对当前的 GWAS 数据进行全基因组两位点交互分析，但是由于搜索空间的大小随着相关联的位点个数的增加而指数上升，BOOST 很难扩展为大于 2 个位点的交互分析。

随机算法[112, 114, 118]并不枚举所有可能的 *m* 个位点的组合，而是采用随机采样过程搜索多位点交互空间，其中 BEAM (*Bayesian epistasis association mapping*) [118] 是其中的一个代表。BEAM 的输入数据是病例对照（case-control）基因型数据，反复迭代使用马尔科夫链蒙特卡罗方法 (MCMC)计算一个位点和疾病相关或和其他位点有交互作用的后验概率。Tang 等[114]对 BEAM 进一步扩展，提出了 epiMODE（epistatic module detection）方法。epiMODE 采用吉布斯（Gibbs）取样和反向（reversible jump）MCMC 过程搜索显著的多位点交互模块。

数据挖掘和机器学习方法采用诸如神经网络[121]、随机森林[122]、自引导（boosting)[112]、预测规则学习[116]等方法来搜索具有统计显著性的多位点交互。大多数数据挖掘和机器学习方法采用启发式规则避免穷尽搜索，例如 SNPruler[116]首先采用规则搜索算法获得可能的交互然后再采用卡方统计度量这些交互的显著性。步进搜索方法首先根据某些单位点统计方法从所有的 SNP 位点中选择出一个子集，然后再对这个子集里的 SNP 位点进行多位点的交互测试[123,124]。与穷尽搜索方法相比，步进方法通常快很多。当致病基因有一定大小的单位点边沿效应时，步进方法在探测致病基因上也有相当不错的表现。但是步进方法对没有单位点边沿效应或单位点边沿效应较小的多位点交互缺乏探测能力，同样基于随机搜索、机器学习等的启发式算法不能保证发现所有显著的多位点交互。

本章提出了一个基于相对频繁项的多位点交互探测算法（Epistasis Detector based on the Clustering of relatively Frequent items，EDCF）。EDCF 采用穷尽搜索方法搜索两位点的交互模块，然后在此基础上采用步进的方法进行高阶交互的寻找。EDCF 把所有的基因型组合聚类成三组，分别代表发病组频繁基因型组合、控制组频繁基因型组合和其他基因型组合，聚类的统计显著性的大小用皮尔逊卡方统计（Pearson

χ^2）来评估。大量模拟测试结果显示 EDCF 比 MB-MDR、BOOST、SNPRuler 和 epiMODE 能更快更强地发现多位点交互模块。通过对一组真实的视网膜黄斑部退化（age-related macular degeneration , AMD）全基因组病例数据集的测试，EDCF 发现了一些在患病组（case）中显著频繁的基因型组合。

4.2 定义和符号

一个 SNP 位点上的基因型可以编码为 0、1 或 2，0 表示该 SNP 位点上两条染色体的取值是群体中多数取值；1 表示该 SNP 位点上一条染色体的取值是群体中多数取值，另一条为群体中少数取值；2 表示表示该 SNP 位点上两条染色体的取值是群体中少数取值。例如在 SNP 位点 A 上假设人群多数取值为 *A*，少数取值为 *a*，则基因型 *A/A*、*A/a* 和 *a/a* 分别编码为 0、1 和 2。在病例对照全基因组关联分析研究中，输入数据是 *N* 个个体的二值疾病状态及他们在 *M* 个 SNP 位点上的基因型。我们用 *S* 表示 *M* 个 SNP 位点的有序集合，s_i 表示 *S* 中的第 *i* 个 SNP 位点，$f_m(s_i)$表示在 s_i 上少数取值在人群中的出现率，$g_i(j)$表示第 *j* 个个体在 s_i 上的基因型。

N^a 和 N^u 分别表示数据集中病例组（case）和对照组（control）人数。$(v_1, \dots, v_d)$ 表示在 SNP 位点 $s_{i_1}, \dots, s_{i_d}$ 上的 *d*-位点基因型组合，$n^a_{v_1,\dots,v_d}$ 和 $n^u_{v_1,\dots,v_d}$ 分别表示病例组和对照组在 $s_{i_1}, \dots, s_{i_d}$ 上的基因型组合为$(v_1, \dots, v_d)$的人数，让 $n^t_{v_1,\dots,v_d} = n^a_{v_1,\dots,v_d} + n^u_{v_1,\dots,v_d}$。令 $f_{v_1,\dots,v_d}$ 表示 *d*-位点基因型组合 $(v_1, \dots, v_d)$在人群中出现率，在连锁平衡的情况下，$f_{v_1,\dots,v_d}$ 可以由各位点基因型的出现率计算得出；令 $p_{v_1,\dots,v_d}$ 为$(v_1, \dots, v_d)$的发病率（在 SNP 位点 $s_{i_1}, \dots, s_{i_d}$ 上的 *d*-位点基因型组合为$(v_1, \dots, v_d)$的个体发病的概率），那整个人群的疾病发病率 *p* 为：

$$p = \sum_{v_1,\dots,v_d=0,1,2} f_{v_1,\dots,v_d} p_{v_1,\dots,v_d} \qquad (4.1)$$

4.3 基因型组合聚类

关联分析的主要目标是发现基因型值的分布在病例组中与对照组中有显著差异的 SNP 位点。相关基因变异导致疾病发生概率的大小、基因型在人群中的分布频率、遗传标记与致病基因的连锁不平衡情况及实验时的采样错误均会对基因型在病例组与对照组的分布差异有影响。GWAS 隐含假设在疾病相关位点，不同基因型组合的发病率显著不同，如表 4.1 所示，在 SNP 位点 A 和 B，基因型组合 *aaBB*、*aabb* 和

AABb 的发病率明显高于 *AABB* 和 *aaBb*。发病率高于人群疾病发病率的基因型组合认为是高风险组合，如表 4.1 中背景为灰色字体为粗体的那些单元；其他的组合认为是低风险组合。可是我们不能直接利用该假设，因为真正的致病基因或其相关联的 SNP 位点和疾病模型发病率表是未知的。实际上在病例对照研究中我们只能观察到基因型组合在病例组与对照组中的分布情况，观察结果通常用如表 4.2 所示的列联表表示。

表 4.1 2 位点交互作用疾病模型的发病率表。 其中 $f_m(\mathrm{A}) = f_m(\mathrm{B}) = 0.4$，人群的疾病发病率 $p = 0.024$。背景为灰色字体为粗体的那些单元认为是高风险组合，其他的组合认为是低风险组合。

		SNP B		
		BB	*Bb*	*bb*
SNP A	*AA*	0.0142	**0.0321**	0.022
	Aa	0.022	**0.0254**	**0.0245**
	aa	**0.0448**	0.0025	**0.0424**

表 4.2 来自表 4.1 所示人群的一个随机取样的 2 位点交互作用的列联表。表格各个单元中，括号中是对照组的人数，括号外是病例组的人数。该病例对照组研究中，病例组和对照组的人数相等，为 $N^a = N^u = 400$.

		SNP B		
		BB	*Bb*	*bb*
SNP A	*AA*	40(53)	**74(61)**	**20(18)**
	Aa	**62(60)**	102(105)	27(30)
	aa	**53(18)**	7(45)	**15(10)**

一个常用的算法 MDR[25]采用了一种简单的策略来定义高风险与低风险基因型组合：计算列联表中各单元病例组与对照组的人数比，如果高于病例组总人数和对照组总人数之比（即 N^a/N^u），则对应的基因型组合是高风险的，否则是低风险的。在表 4.2 中，MDR 认为 *AABb*、*AAbb*、*AaBB*、*aaBB* 和 *aabb*（即灰色背景粗体的单元对应的基因型组合）是高风险的。然而实际的 GWAS 中的采样存在着随机采样错误，这种简单的策略实际上有时不能正常工作，特别是当有些单元的病例组与对照组的人数比与 N^a/N^u 接近时，例如，表 4.2 中 *AAbb* 和 *AaBB* 被错误地认为是高风险基因型组合，而 *AaBb* 被错误地认为是低风险基因型组合。

为了克服上述缺点，本章提出把所有的基因型聚类成三组：G_0、G_1 和 G_2，其中 G_0 表示在病例组中显著频繁的基因型组合的聚类（即被假定为高风险基因型组合），

G_2 表示在对照组中显著频繁的基因型组合的聚类（即被假定为低风险基因型组合），G_1 是其余基因型组合的聚类。

为了进行基因型聚类，我们首先需要衡量基因型组合的相对频繁程度，并计算其统计有效性。考虑一个在 d 个 SNP 位点 $s_{i_1}, \ldots, s_{i_d}$ 上的基因型组合$(v_1, \ldots, v_d)$，空假

设是该组合与疾病无关，即该组合的疾病发病率 $p_{v_1,\ldots,v_d}$ 和人群的疾病发病率是没有差异的，那在列联表对应基因型组合$(v_1, \ldots, v_d)$的单元中病例组的人数应服从一个二项式分布，即：$n^a \sim B(n = n^t;\ p_a = N^a/(N^a + N^u))$ (在此及以后的表达式中,为了行文简洁，在不影响语义表达的情况下，记号的下标 $v_1, \ldots, v_d$ 均被省略)。同理，在空假设的成立的前提下，列联表对应基因型组合$(v_1, \ldots, v_d)$的单元中对照组的人数 $n^u \sim B(n = n^t;\ p_u = N^u/(N^a + N^u))$。对于一个给定的显著性水平 α_s（例如 $\alpha_s = 0.05$），定义 T_a 为病例组对应显著性水平 α_s 的人数临界值，即 $Pr(k > T_a \mid n^t, p_a) < \alpha_s$；同样定义 T_u 为对照组对应显著性水平 α_s 的人数临界值，即 $Pr(k > T_u \mid n^t, p_u) < \alpha_s$。

定义 4.1 如果 $n^a > T_a$, $(v_1, \ldots, v_d)$是一个在显著性水平 α_s 下病例组中的相对频繁项；如果 $n^u > T_u$, $(v_1, \ldots, v_d)$是一个在显著性水平 α_s 下对照组中的相对频繁项。

定义 4.2 给定一个显著性水平 α_s 和一组 SNP 位点，G_0 定义为在病例组中相对频繁项的集合；G_2 定义为在对照组中相对频繁项的集合；G_1 定义为其他基因型组合的集合。

4.4 多位点交互统计显著性计算

一旦在某 d 个 SNP 位点 $s_{i_1}, \ldots, s_{i_d}$ 上所有的基因型组合聚类成 G_0、G_1 和 G_2，我们很容易得到如表 4.3 所示的 3 行×2 列的列联表，该表的 3 行分别代表 G_0、G_1 和 G_2，2 列分别代表病例组和对照组，该表在行“G_i” ($i = 0, 1,$ 或 2)和列“Cases” (或“Controls”)上的单元表示在病例组（或对照组）中带 G_i 中的基因型组合的人数。这个列联表上的自由度为 2 的卡方测试[125]（这里记做 $X_2^2(i_1,\ldots,i_d)$ ）就能作为相对频繁项聚类的统计显著性的度量。按照直觉，高的 $X_2^2(i_1,\ldots,i_d)$ 意味着 $s_{i_1}, \ldots, s_{i_d}$ 是一组交互的 SNP 位点，可是，这不完全正确，因为在这些位点中有些可能是冗余的（如某些位点对列联表的各个单元没有影响）。考虑这种情况，下面定义最小的可能交互 SNP 位点子集为交互模块。

定义 4.3 在下面条件满足的情况下，$(s_{i_1}, \ldots, s_{i_d})$是一个在显著性水平 α 下的一个交互模块：

（1） $X_2^2(i_1,\ldots,i_d)$ 的 p-value 小于或等于 α；

（2）不存在$(i_1, \ldots, i_d)$的一个真子集$(i'_1, \ldots, i'_{d'})$ （$d' < d$）使得： $X_2^2(\ i'_1, \ldots, i'_{d'}) \geq$

$X_2^2(i_1,...,i_d)$.

表 4.3 一个 3×2 的列联表

	病例组（Cases）	**对照组（controls）**
G_0	$\sum_{(v_1,...,v_d)\in G_0} n^a_{v_1,...,v_d}$	$\sum_{(v_1,...,v_d)\in G_0} n^u_{v_1,...,v_d}$
G_1	$\sum_{(v_1,...,v_d)\in G_1} n^a_{v_1,...,v_d}$	$\sum_{(v_1,...,v_d)\in G_1} n^u_{v_1,...,v_d}$
G_2	$\sum_{(v_1,...,v_d)\in G_2} n^a_{v_1,...,v_d}$	$\sum_{(v_1,...,v_d)\in G_2} n^u_{v_1,...,v_d}$

当有很多交互模块是，考虑到生物学家可能只对前 k 个最显著的交互模块感兴趣，对于给定的 d，我们在下节提出算法 EDCF（Epistasis Detector based on the Clustering of relatively Frequent items）搜索 top-k 个最显著的基因组多位点交互模块。

4.5 EDCF 算法

EDCF 是迭代搜索 top-k 个最显著的基因组 d 位点交互模块。在当前的 GWAS 中，SNP 位点的数量从几十万到几百万，对于 $d \geq 3$ 位点的交互，穷尽搜索整个交互空间是不切实际的。假定 d 位点交互模块其中某个子集的交互作用也具有一定的统计显著性，为了发现 top-k 个最显著的基因组 d 位点交互模块，我们先要获得 top-kf_s 个最显著的基因组 d-1 个位点交互模块，其中 $f_s \geq 1$ 为一个放大因子。采用有效地实现，在当前的 GWAS 中枚举出每一对 SNP 位点是可行的，因此当搜索 2 位点的交互时上述递归过程终止。

为了发现 2 位点交互模块，我们对每对可能的 SNP 位点(s_{i_1}, s_{i_2})计算 $X_2^2(i_1, i_2)$。一旦 2 位点交互模块通过穷尽搜索被找到，对满足下面条件的 SNP 位点 3 元组(i_1, i_2, i_3)计算 $X_2^2(i_1, i_2, i_3)$: (s_{i_1}, s_{i_2})是一个 2 位点交互模块，且 $s_{i_3} \neq s_{i_1}, s_{i_2}$。上述递推过程继续直到找到 top-$k$ 个最显著的基因组 d 位点交互模块。

计算 X_2^2 的详细过程如下。

采用位操作可以加快列联表的计数，我们采用同 Wan *et al.*, 2010c 相同的方法用位向量（bit vector）来表示数据集中的基因型，用 $6M$ 个位向量来表示 N 个体（其中病例组 N^a 个个体，对照组 N^u 个个体）M 个 SNP 位点的基因型数据。每一个 SNP

位点 s_i 病例组的基因型数据用 3 个位向量 A_0^i、A_1^i 和 A_2^i 表示，对照组的基因型数据也用 3 个位向量 U_0^i、U_1^i 和 U_2^i 表示。A_v^i（v = 0, 1 或 2）有 N^a 位，而 U_v^i（v = 0, 1 或 2）有 N^u 位，当病例组（或对照组）第 j 个个体在 SNP 位点 s_i 上的基因型值为 v 时，A_v^i（或 U_v^i）的第 j 位被置为 1，否则被置为 0。

考虑 $s_{i_1}, \ldots, s_{i_d}$ 上的一个 d-位点基因型组合$(v_1, \ldots, v_d)$，在病例组和对照组中其 d-位点基因型组合为$(v_1, \ldots, v_d)$的个体个数 $n^a_{v_1, \ldots, v_d}$ 和 $n^u_{v_1, \ldots, v_d}$ 可以下面的位操作计算：

$$n^a_{v_1, \ldots, v_d} = \text{BitCount}\,(A^{i_1}_{v_1} \& \ldots \& A^{i_d}_{v_d}) \tag{4.2}$$

$$n^u_{v_1, \ldots, v_d} = \text{BitCount}\,(U^{i_1}_{v_1} \& \ldots \& U^{i_d}_{v_d}) \tag{4.3}$$

其中&是一个位与操作符，BitCount 是一个对位向量中为 1 的位数进行计数的函数，在 EDCF 算法中，该操作频繁使用，为了节约计数的时间，一个简单而快速的方法被采用：构造一个 hash 表映射 16 位 2 进制数到该数对应的位向量中的为 1 的位数，BitCount 采用查 hash 表的方式实现。

一旦 $n^a_{v_1, \ldots, v_d}$ 和 $n^u_{v_1, \ldots, v_d}$ 被计算出来，$(v_1, \ldots, v_d)$就按照 4.3 所示的方法分配到 G_0、G_1 或 G_2。当所有 $(v_1, \ldots, v_d)$聚类成 G_0、G_1 和 G_2，表 4.3 所示的列联表就构造出来了，进而自由度为 2 的卡方统计量 $X_2^2(i_1, \ldots, i_d)$就很容易计算。

如前所述，EDCF 是一个递归算法，为了获得 top-k 个最显著的基因组 d 位点交互模块，EDCF 先要获得 top-kf_s 个最显著的基因组 d-1 个位点交互模块，这意味着 EDCF 首先搜索 top-$k f_s^{d-2}$ 最显著的 2 位点交互模块，再寻找 top-$k f_s^{d-3}$ 最显著的 3 位点交互模块，上述过程反复进行直至 d 位点交互模块。对每个 $d' \in \{2, 3, \ldots, d\}$, EDCF 维护一个小根堆 H 存储候选的 d'位点交互模块，一个集合 $Q_{d'}$ 存储在显著性水平 α 下的 d'位点交互模块。堆 H 中的每个元素包含 $X_2^2(i_1, \ldots, i_{d'})$和对应的 d' SNP 位点信息 $(i_1, \ldots, i_{d'})$，堆中每个元素的值比它的两个孩子都小，用来比较的元素大小的属性有 $X_2^2(i_1, \ldots, i_{d'})$、$i_1$、…、$i_{d'}$。堆 H 根的 X_2^2 值用 $H.r.X_2^2$ 表示，H 中元素个数为 $k f_s^{d-d'}$，堆最多储存 $k f_s^{d-d'}$ 个元素。当一个新的元素$(i_1, \ldots, i_{d'})$到达且 H 有空余的空间，H 接受新元素并做相应调整以维持小根堆的性质；当 H 已满，如果新的元素的 X_2^2 值小于 $H.r.X_2^2$，则新元素被丢弃，否则根元素被新元素替换，H 进行相应调整以维持小根堆的性质，上述操作记为 H.insert($(i_1, \ldots, i_{d'})$, X_2^2)。

如果我们已经获得了 top-kf_s 个最显著的基因组 d-1 个位点交互模块，这些模块储存在堆 H'中，且对所有 d' = 2, 3, ..., d-1，$Q_{d'}$ 也已知，那 EDCF 就采用如下方法搜索 top-k 个最显著的基因组 d 个位点交互模块并把它们储存在堆 H (H 才开始为空)中：对 H'中的每一个 $d-1$ 元组(i_1, ..., $i_{d\text{-}1}$)，EDCF 为每个可能的 $i_d \neq i_1$, ..., $i_{d\text{-}1}$ 计算 X_2^2(i_1, ..., $i_{d\text{-}1}$, i_d)，进而和 H 的根节点及在 $Q_{d'}$ ($1 < d' < d$)中 (i_1, ..., $i_{d\text{-}1}$)的所有真子集进行比较，如果 X_2^2(i_1, ..., $i_{d\text{-}1}$, i_d)的值大于 $H.r.X_2^2$，且没有发现其真子集的 X_2^2 值大于或等于 X_2^2(i_1, ..., $i_{d\text{-}1}$, i_d)，则实施堆插入操作 H.insert((i_1, ..., $i_{d\text{-}1}$, i_d), X_2^2)，最后 H 中保留的就是 top-k 个最显著的基因组 d 个位点交互模块。EDCF 的详细步骤请参考图 4.1。

EDCF的空间复杂度为$O(NM + (d\text{-}1)k f_s^{d-2} + (d\text{-}2)|Q|)$，其中$|Q|$表示最大的$Q_{d'}$ ($1 < d' < d$)中包含的元素个数。EDCF的时间复杂度分析如下：Step 1.2对所有可能的i计算$T_a(i, \alpha_s)$和$T_u(i, \alpha_s)$所需时间为$O(N)$。函数DetectInteraction对M个SNP位点中所有的SNP对相关的2位点基因型组合聚类成G_0、G_1和G_2并计算出卡方统计值 X_2^2 所用的时间为$O(NM^2)$。把一个元素插入到大小为k的堆中所需时间为$O(\log k)$，当$d = 2$时，EDCF的时间复杂度为$O(N + (\log k + N)M^2)$，因为和N与M相比k相当小，因此算法的时间复杂度近似为$O(NM^2)$；当$d = 3$时，EDCF需$O((\log(kf_s) + N)M^2)$获得top- kf_s个最显著的基因组2位点交互模块，假设位计数函数是高效的且其时间为常数，则把2位点交互模块扩展为3位点交互模块需$O((\log k + |Q|)kf_sM)$的时间，这样EDCF的时间复杂度为$O(N + (\log k + N)M^2 + (\log k + |Q|)kf_sM)$。当$d \geq 4$时，EDCF的时间复杂度为$O((\log(k f_s^{d-2}) + N)M^2 + (\log(k f_s^{d-3}) + |Q|)(d-2)\, k\, f_s^{d-2} M + k2^d)$，其中最后一项$k2^d$是Step 3.2所需的时间。

4.6 假阳性错误控制

在GWAS多位点交互分析中的一个需要克服的关键问题是在多重测试中如何控制假阳性错误（即类型I错误），排列检验(Permutation test)和Bonferroni纠正是常用的处理多重测试问题(multiple testing problem)的两个常用的方法。EDCF有两个层次的多重比较：第一层次的多重比较在于M个SNP位点中不同的d位点组合为$\binom{M}{d}$个；第二层次的多重比较在于如下事实：EDCF把3^d可能的基因型组合聚合成3类然后进行自由度为2的卡方测试，当d增大时，即使数据是随机生成的，卡方值也会由于非随机的聚类而明显增大。理论上把3^d可能的基因型分成3类有3^{3^d}种方式，考虑到实际上聚类的方

式远低于3^{3^d}，为处理EDCF的多重测试问题使用简单的Bonferroni纠正（用做阈值纠正后的显著性水平α等于纠正前的显著性水平α_0除以$\binom{M}{d}3^{3^d}$）过于保守，可当M较大时采用排列检验由于所需的计算资源过大而不切实际。

为了有效地控制 EDCF 的假阳性率（类型 I 错误率），我们提出结合排列检验和 Bonferroni 纠正的处理多重测试问题方法：考虑到 EDCF 两层次的多重比较是独立的，我们采用 Bonferroni 纠正控制第一层次的多重比较，采用排列检验控制第二层次的多重比较，具体地说，用做阈值的显著性水平 α 定义为：

$$\alpha = \alpha_0 / \binom{M}{d} \tag{4.4}$$

其中 $\binom{M}{d}$ 是采用 Bonferroni 纠正控制 d 位点组合引起的多重比较，而 α_0 是采用排列检验在较小数据集上采用排列检验获得，对于不同的 d，α_0 的取值不同，目标是由式(4.4)计算得到显著性水平 α 能控制测试的假阳性率在 0.05 之内。

INPUT: Genotypes of M SNPs and disease status data of N individuals (N^a affected and N^u unaffected); α_s, α, k, f_s, and d.
OUTPUT: The top-k significant d-locus interaction modules under the significance level α.

1. **initialization**
1.1. convert the input data into bit vectors: $A_0^1, A_1^1, A_2^1, \ldots, A_0^M, A_1^M, A_2^M, U_0^1, U_1^1, U_2^1, \ldots, U_0^M, U_1^M, U_2^M$;
1.2. precalculate $T_a(i, \alpha_s)$ and $T_u(i, \alpha_s)$ for every $i = 1, \ldots, N$;
1.3. calculate $X_2^2(i)$ for each SNP $i = 1, \cdots, M$;
2. call Procedure **DetectInteraction**(d, k), which returns a heap H;
3. **for** each module $(i_1, \ldots, i_d)$ in H, **if** the p-value of its X_2^2 is not greater than α **then**
3.1 **if** $d < 4$ **then** output $(i_1, \ldots, i_d)$;
3.2 **else if** there is no such subset $(i'_1, \ldots, i'_{d'})(1 < d' < d)$ of $(i_1, \cdots, i_d)$ such that $X_2^2(i'_1, \cdots, i'_{d'}) \geq X_2^2(i_1, \cdots, i_d)$ **then** output $(i_1, \cdots, i_d)$.

PROCEDURE DetectInteraction(d, k)
(1) initiate a heap H of size k and set $T_x = 0$;
(2) **if** $d = 2$, **then**
(2.1) **for** each SNP pair (s_{i_1}, s_{i_2}) $(1 \leq s_{i_1} < s_{i_2} \leq M)$ **do**
(2.1.1) $n_{00} = n_{01} = n_{10} = n_{11} = n_{20} = n_{21} = 0$;
(2.1.2) **for** each genotype combination (v_1, v_2) **do**
$n^a = \text{BitCount}(A_{v_1}^{i_1} \& A_{v_2}^{i_2})$; $n^u = \text{BitCount}(U_{v_1}^{i_1} \& U_{v_2}^{i_2})$;
if $n^a \geq T_a(n^a + n^u)$ **then** $n_{00} = n_{00} + n^a$; $n_{01} = n_{01} + n^u$;
else if $n^u \geq T_u(n^a + n^u)$ **then** $n_{20} = n_{20} + n^a$; $n_{21} = n_{21} + n^u$;
else $n_{10} = n_{10} + n^a$; $n_{11} = n_{11} + n^u$;
calculate $X_2^2(i_1, i_2)$ for each pair;
if $X_2^2(i_1, i_2) > X_2^2(i_1)$ and $X_2^2(i_1, i_2) > X_2^2(i_2)$ and $X_2^2(i_1, i_2) > T_x$ **then**
H.insert$((i_1, i_2), X_2^2)$;
if H is full **then** $T_x = H.r.X_2^2$;
(2.2) return H;
(3) **else if** $d > 2$, **then**
(3.1) heap $H' = \text{DetectInteraction}(d - 1, k \times f_s)$;
(3.2) sort the elements of H' in descending order;
(3.3) put the elements of H' whose corresponding p-value is not greater than α into set Q_{d-1};
(3.4) **for** each $(i_1, \ldots, i_{d-1})$ in H' (according to the order) **do**
for each $i_d : i_d \neq i_1, \ldots, i_{d-1}, 1 \leq i_d \leq M$ **do**
calculate $X_2^2(i_1, \ldots, i_d)$;
if $X_2^2(i_1, \ldots, i_d) > X_2^2(i_1, \ldots, i_{d-1}), X_2^2(i_d)$ and T_x **then**
for each $(i'_1, \ldots, i'_{d'}) \in Q_{d'} (1 < d' < d)$ **do**
if $(i'_1, \ldots, i'_{d'}) \subset (i_1, \ldots, i_d)$ and $X_2^2(i'_1, \ldots, i'_{d'}) \geq X_2^2(i'_1, \ldots, i'_{d'})$ **then** jump to (3.4);
H.insert$((i_1, \ldots, i_d), X_2^2)$;
if H is full **then** $T_x = H.r.X_2^2$;
(3.5) free the memory of H' and return H;

图 4.1 EDCF 算法

4.6 实验设计

为了评估 EDCF 的有效性，我们使用不同的疾病模型进行了大量的模拟测试。模拟测试对 EDCF 和最近提出的 GWAS 多基因交互分析算法 MB-MDR[119], BOOST [117], SNPruler[116], epiMODE[114]及简单的 2-位点卡方测试 (ChiSQ)进行了性能比较。模拟测试数据是采用以前文献如[117]等所阐述的方法生成。对于每个背景 SNP 位点 s_i，少数值频度 $f_m(i)$在[0.05, 0.5]范围内随机取值，然后按照 $f_m(i)$的值根据 Hardy-Weinberg 平衡原则生成该位点上的基因型值。最后模拟数据生成其器按照疾病模型生成病例组和对照组与疾病相关 SNP 位点上的基因型数据，具体生成方法见 4.6.1 节。

4.6.1 疾病模型

本章模拟测试考虑2类疾病模型有单位点边际效应（marginal effect）的多位点交互模型（epistasis model）和没有单位点边际效应的多位点交互模型，交互的位点有2个或3个。多基因交互疾病模型通常用多个SNP位点上对应基因型组合的发病概率(penetrance)或发病奇异率(odds)列联表表示。对于一个d-位点基因型组合为(v_1, ... , v_d)，其发病奇异率定义为：

$$r_o(v_1,...,v_d)=\frac{p_{v_1,...,v_d}}{1-p_{v_1,...,v_d}}. \tag{4.5}$$

理论上一个基因型组合的发病率可以任意赋值，可是实际上大多数研究者只使用一些受限的疾病模型，在很多情况下，疾病模型只有两个自由的参数，一个表示基效应（baseline effect，如表4.4中的β），另一个表示附加效应（additional effect, 如表4.4中的θ）。虽然疾病模型的各基因型组合的发病率可直接由这两个参数决定，可实际

表 4.4 有单位点边缘效应的 2-位点交互疾病模型

Model 1		SNP B		
		BB	*Bb*	*bb*
SNP	*AA*	β	β	β
	Aa	β	$\beta(1+\theta)$	$\beta(1+\theta)^2$
	aa	β	$\beta(1+\theta)^2$	$\beta(1+\theta)^4$

Model 2		SNP B		
		BB	*Bb*	*bb*
SNP	*AA*	β	$\beta(1+\theta)$	$\beta(1+\theta)$
	Aa	$\beta(1+\theta)$	β	β
	aa	$\beta(1+\theta)$	β	β

Model 3		SNP B		
		BB	*Bb*	*bb*
SNP	*AA*	β	β	$\beta(1+\theta)$
	Aa	β	$\beta(1+\theta)$	β
	aa	$\beta(1+\theta)$	$\beta(1+\theta)$	β

Model 4		SNP B		
		BB	*Bb*	*bb*
SNP	*AA*	β	$\beta(1+\theta)$	β
	Aa	$\beta(1+\theta)$	β	$\beta(1+\theta)$
	aa	β	$\beta(1+\theta)$	β

上

研究者通常指定人口发病率 p 和另外一个参数（遗传可能性 genetic heritability 或边缘效应 marginal effect），然后用数值计算方法确定这两个参数。

遗传可能性（genetic heritability）的大小定义为(Culverhouse et al., 2002)：

$$h^2 = \frac{\sum_{v_1,\ldots,v_d=0,1,2} (p - p_{v_1,\ldots,v_d})^2 f_{v_1,\ldots,v_d}}{p(1-p)} \tag{4.6}$$

按照 Zhang and Liu (2007)，一个致病位点的边缘效应（marginal effect）的大小定义为：

$$\lambda = \frac{p_{Aa} / p_{AA}}{(1-p_{Aa})/(1-p_{AA})} - 1 \tag{4.7}$$

一旦通过数值计算方法由人口发病率 p、疾病相关的 SNP 位点少数值的频度 f_m、h^2 或 λ 确定发病率或奇异率列联表，这些位点在疾病组和对照组中基因型组合的条件分布就能计算出来，进而模拟数据生成器按照条件分布随机生成这些位点的基因型。

对于有单位点边缘效应的 2-位点交互，采用的是表 4.4 所示的 4 个疾病模型。模型 1 1 是两个位点的交互的乘法模型(123)，模型 2 对应 Neuman 和 Rice（Neuman and Rice, 1992）研究的 2 位点交互疾病模型中的 Ep-6，模型 3 是 Li 和 Reich 论文中的模型 M86 (Li and Reich, 2000)，而模型 4 是常用的 XOR 模型。这些模型同样也被用于以前的研究中 Wan et al., 2010c。当疾病相关的 SNP 少数值的频度 f_m 固定不变时，模型 1-4 的参数 β 和 θ 通过数值方法由人口发病率 p 和遗传可能性 h^2 确定。

对于没有单位点边缘效应的 2-位点交互，我们同 Wan 等[116]一样采用表 4.5 和 4.6 所示的 60 个 2-位点纯交互模型（pure epistasis model）。

对于 3-位点的交互模型，我们采用一个有单位点边缘效应的疾病模型（表 4.7 中的 Model 5）和一个没有边缘效应的疾病模型（表 4.7 中的 Model 6）。模型 5 同 Zhang and Liu (2007)中的 Model 4 本质上是一样的，为了和 Zhang and Liu (2007)保持一致，我们采用单位点边缘效应 λ 控制模型中的参数。模型 6 是 Culverhouse et al. (2002)提出的一个具有最大遗传可能性但没有单位点边缘效应的模型，模型 6 要求 SNP 的少数值频度为 0.5，人口疾病发病率 $p \in (0, 1/16]$。

为了进一步评估分子标记和致病 SNP 之间的连锁不平衡（linkage disequilibrium，LD）对算法的探测能力的影响，植入疾病相关的 SNP 位点时采用下面两种不同的方法：直接植入致病 SNP 位点，或不直接致病 SNP 位点而是植入和致病 SNP 位点连锁

不平衡(LD: r^2 = 0:7)的 SNP 位点。

4.6.2 统计有效性

在比较不同算法的性能时，我们采用 Wan et al. (2010b,c)使用的度量标准探测能力（discrimination power）：power = n_c ／n，其中 n 为重复测试的次数，n_c 为在所有重复测试中算法成功探测出植入的 SNP 位点的测试次数。除 epiMODE 外，如果植入的 SNP 位点集合是算法发现的所有交互模块中统计显著性最强，且对应的 p-value 小于预定的阈值，则算法成功探测出植入的 SNP 位点。epiMODE 并不输出交互模块的统计显著性水平，因此如果它输出的某个模块中包含且只包含植入的 SNP 位点，则 epiMODE 被认为成功探测出了植入位点。在进行 d-位点交互的疾病模型测试时，我们比较能输出 d-位点交互模块的算法的性能。在测试中，模拟数据集为平衡设计即病

表 4.5 无单位点边缘效应的 2-位点纯交互疾病模型（Model 1-30）

Model	*AABB*	*AaBB*	*aaBB*	*AABb*	*AaBb*	*aaBb*	*AAbb*	*Aabb*	*aabb*	f_m	h^2
1	0.486	0.96	0.538	0.947	0.004	0.811	0.64	0.606	0.909		
2	0.469	0.956	0.697	0.945	0.019	0.585	0.786	0.407	0.013		
3	0.498	0.954	0.786	0.978	0.038	0.428	0.59	0.821	0.38	0.2	
4	0.505	0.988	0.624	0.945	0.085	0.807	0.969	0.116	0.159		
5	0.486	0.963	0.512	0.941	0.006	0.899	0.691	0.541	0.614		0.4
6	0.077	0.656	0.88	0.892	0.235	0.312	0.174	0.842	0.106		
7	0.895	0.323	0.161	0.068	0.728	0.806	0.925	0.233	0.362		
8	0.805	0.251	0.085	0.002	0.668	0.638	0.83	0.079	0.542	0.4	
9	0.307	0.682	0.958	0.997	0.39	0.281	0.012	0.99	0.698		
10	0.083	0.891	0.037	0.619	0.271	0.691	0.853	0.079	0.742		
11	0.5	0.926	0.615	0.895	0.131	0.647	0.858	0.16	0.999		
12	0.413	0.851	0.535	0.831	0.008	0.58	0.692	0.268	0.736		
13	0.455	0.848	0.897	0.89	0.088	0.016	0.562	0.686	0.467	0.2	
14	0.609	0.98	0.98	0.993	0.3	0.275	0.876	0.483	0.683		
15	0.446	0.844	0.774	0.879	0.044	0.233	0.492	0.796	0.41		0.3
16	0.891	0.362	0.48	0.213	0.829	0.601	0.925	0.267	0.685		
17	0.077	0.689	0.417	0.763	0.15	0.491	0.196	0.657	0.247		
18	0.132	0.793	0.274	0.799	0.213	0.514	0.255	0.528	0.793	0.4	
19	0.611	0.104	0.759	0.18	0.674	0.019	0.532	0.189	0.681		
20	0.091	0.827	0.863	0.869	0.393	0.415	0.738	0.508	0.363		
21	0.428	0.757	0.812	0.788	0.132	0.044	0.559	0.548	0.373		
22	0.507	0.842	0.605	0.845	0.162	0.629	0.581	0.678	0.729		
23	0.577	0.247	0.428	0.227	0.928	0.578	0.586	0.262	0.158	0.2	
24	0.34	0.637	0.654	0.689	0.017	0.041	0.242	0.866	0.403		
25	0.387	0.726	0.734	0.749	0.09	0.034	0.551	0.401	0.724		0.2
26	0.356	0.891	0.809	0.955	0.508	0.611	0.617	0.755	0.63		
27	0.086	0.536	0.641	0.677	0.275	0.096	0.219	0.413	0.712		
28	0.855	0.339	0.772	0.513	0.651	0.607	0.25	0.999	0.154	0.4	
29	0.506	0.838	0.024	0.603	0.454	0.957	0.729	0.427	0.753		
30	0.393	0.764	0.664	0.85	0.398	0.733	0.406	0.927	0.147		

例组中的人数和对照组中的人数相等（$N^a = N^u$）。对于 EDCF，除非另外说明，参数 $\alpha_s = 0{:}05$，$k = 20$，$f_s = M / k$ 。

表 4.6 无单位点边缘效应的 2-位点纯交互疾病模型（Model 31-60）

Model	*AABB*	*AaBB*	*aaBB*	*AABb*	*AaBb*	*aaBb*	*AAbb*	*Aabb*	*aabb*	f_m	h^2
31	0.463	0.703	0.431	0.653	0.277	0.806	0.83	0.008	0.129		
32	0.319	0.507	0.569	0.553	0.105	0.045	0.203	0.777	0.28		
33	0.627	0.393	0.335	0.396	0.779	0.953	0.174	0.842	0.106	0.2	
34	0.297	0.54	0.441	0.541	0.072	0.278	0.434	0.293	0.228		
35	0.332	0.562	0.573	0.583	0.112	0.147	0.399	0.496	0.033		0.1
36	0.137	0.484	0.187	0.482	0.166	0.365	0.193	0.361	0.43		
37	0.469	0.198	0.754	0.337	0.502	0.141	0.339	0.453	0.285		
38	0.478	0.311	0.864	0.387	0.579	0.263	0.634	0.436	0.138	0.4	
39	0.068	0.299	0.017	0.289	0.044	0.285	0.048	0.262	0.174		
40	0.539	0.12	0.258	0.165	0.378	0.325	0.123	0.426	0.276		
41	0.492	0.664	0.481	0.642	0.33	0.746	0.656	0.396	0		
42	0.499	0.639	0.765	0.666	0.389	0.083	0.543	0.527	0.953		
43	0.212	0.35	0.116	0.336	0.054	0.495	0.227	0.273	0.495	0.2	
44	0.805	0.683	0.638	0.657	0.936	0.989	0.85	0.564	0.866		
45	0.638	0.488	0.383	0.464	0.765	0.957	0.58	0.562	0.719		0.05
46	0.002	0.155	0.214	0.199	0.071	0.022	0.081	0.122	0.135		
47	0.188	0.02	0.171	0.032	0.174	0.059	0.134	0.087	0.092		
48	0.005	0.179	0.251	0.211	0.1	0.026	0.156	0.098	0.156	0.4	
49	0.174	0.321	0.154	0.223	0.254	0.245	0.448	0.025	0.424		
50	0.098	0.219	0.302	0.302	0.126	0.121	0.053	0.308	0.136		
51	0.495	0.415	0.657	0.429	0.616	0.121	0.552	0.331	0.419		
52	0.592	0.691	0.743	0.712	0.493	0.419	0.58	0.746	0.504		
53	0.108	0.194	0.186	0.196	0.037	0.045	0.172	0.073	0.13	0.2	
54	0.112	0.186	0.128	0.193	0.024	0.138	0.079	0.236	0.251		
55	0.272	0.192	0.185	0.172	0.367	0.39	0.345	0.069	0.005		0.025
56	0.166	0.165	0.128	0.114	0.199	0.143	0.281	0.028	0.281		
57	0.108	0.006	0.08	0.026	0.079	0.046	0.021	0.09	0.025		
58	0.006	0.094	0.008	0.079	0.016	0.076	0.052	0.043	0.057	0.4	
59	0.199	0.072	0.168	0.086	0.187	0.076	0.125	0.108	0.226		
60	0.165	0.096	0.262	0.166	0.151	0.091	0.05	0.25	0.056		

4.7 实验结果

在本节没有植入疾病相关 SNP 位点的模拟背景基因型数据首先被用来控制 EDCF 的假阳性率，然后利用不同疾病模型生成的模拟数据被用来测试 EDCF、MB-MDR、BOOST、SNPruler、epiMODE 和 ChiSQ 的性能，最后我们给出 EDCF 在一个真实的 GWAS 数据集上测试的结果。程序 MB-MDR (C++实现), BOOST (64 位), SNPRuler 和 epiMODE 都是从它们的作者的网上下载而来，其中 MB-MDR 是对 MDR

表 4.7 3-位点交互疾病模型

Model 5		SNPs B and C								
		BBCC	*BbCC*	*bbCC*	*BBCc*	*BbCc*	*bbCc*	*BBcc*	*Bbcc*	*bbcc*
SNP A	*AA*	β	β	β	β	β	$\beta(1+\theta)$	β	$\beta(1+\theta)$	β
	Aa	β	β	$\beta(1+\theta)$	β	$\beta(1+l\theta)$	β	$\beta(1+\theta)$	β	β
	aa	β	$\beta(1+\theta)$	β	$\beta(1+\theta)$	β	β	β	β	β

Model 6		SNPs B and C								
		BBCC	*BbCC*	*bbCC*	*BBCc*	*BbCc*	*bbCc*	*BBcc*	*Bbcc*	*bbcc*
SNP A	*AA*	0	0	16p	0	0	0	0	0	0
	Aa	0	0	0	0	4p	0	0	0	0
	aa	0	0	0	0	0	0	16p	0	0

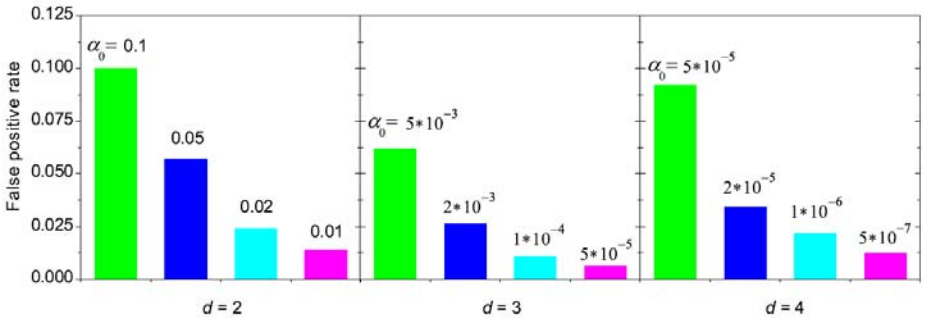
最近的一个扩展，BOOST 是一个搜索 2-位点交互的快速穷尽算法，epiMODE 是 BEAM 最近的一个扩展，SNPRuler 使用规则推断（rule inference）和随机搜索处理 3-位点和高阶交互，ChiSQ 用 C++实现，采用 8 个自由度的 Pearson χ^2 测试探测 2-位点的交互。所有的测试都是在一个 2.8 GHz CPU 和 16 GB RAM 的 64 位 Linux 平台上进行。除非特别申明，对每一个模型和每一个参数集，100 个模拟测试数据集随机生成，每个测试数据集中均包含 2000SNP 位点。

4.7.1 假阳性率

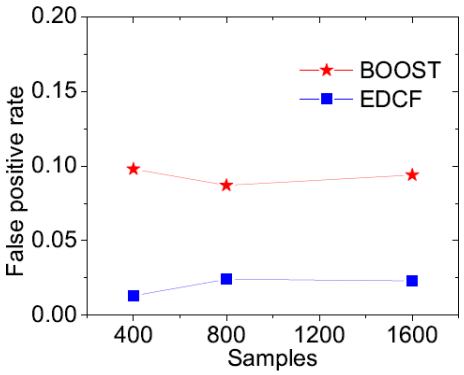
如4.6节所述，EDCF显著性水平阈值设为$\alpha = \alpha_0 / \binom{M}{d}$，我们利用没有植入任何疾病相关位点的模拟基因型数据来测试EDCF在不同的α_0下的假阳性率。在测试时对每一个参数集，1000个模拟数据集随机生成，EDCF的假阳性率定义为$n_f/1000$，其中n_f为EDCF发现交互模块的模拟数据集个数。每一个数据集包含800个个体（病例组400个，对照组400个）在2000SNP（M = 2000）上的基因型信息。图4.2所示的测试结果显示当$d = 2$、3或4时，如果α_0设置成0.02、0.002或0.00002，则EDCF的假阳性率低于0.05，因此在下面的实验中，α_0分别设置成0.02、0.002或0.00002来测试2、3或4位点的交互以控制EDCF的假阳性率。

为了控制 MB-MDR、BOOST、SNPruler 和 epiMODE 的假阳性率，我们采用原文献推荐的方法：MB-MDR 采用排列测试（permutation test）控制假阳性率，它的显著性水平阈值设置为 0.05。BOOST 默认输出所有 $\tau \geq 30$ （τ 的定义请参考 Wan et al., 2010c，$\tau \geq 30$ 对应为多重测试修正前的 p-value $\leq 4.89 \times 10^{-6}$）的 2-位点交互模块。SNPRuler 首先得到 top-k（在测试中 k = 20）个包含 d 个位点的规则，然后输出其中

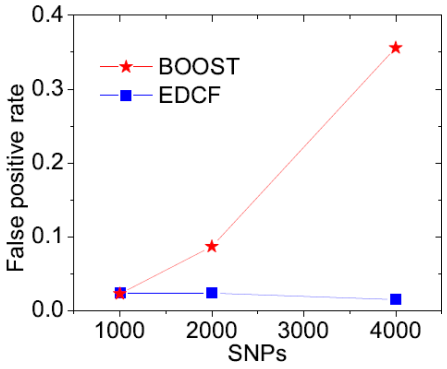
未修正（没有进行多重测试修正）p-value ≤ 1.5×10^{-7} 的规则。ChiSQ 使用 SNPRuler 相同的阈值，而 epiMODE 并不输出其发现的模块的 p-value，没有给用户提供控制显著性水平阈值的接口。



(a) EDCF 的假阳性率（False positive rate）



(b) 数据集中人数发生变化



(c) 数据集的 SNP 位点个数发生变化

图 4.2 EDCF 在背景基因型数据（空模型，没有植入疾病相关位点）下的假阳性率。图(a) 显示 EDCF 在不同的 α_0 和不同的 d 下的假阳性率，图(b)和(c)显示在人数和 SNP 位点数发生变化 时 EDCF 和 BOOST 的假阳性率。

我们使用 1000 个随机生成的背景基因型数据集（每个数据集包含 800 个个体的 2000SNP 位点上的基因型数据，当测试 MB-MDR 时，SNP 位点数减少为 100 以便测试能在 1 天内完成）采用上述的阈值测试各种算法，实验结果显示 EDCF、MB-MDR、BOOST、SNPRuler、epiMODE 和 ChiSQ 的假阳性率分别为 0.024、0.056、0.087、0.028、0.019 和 0.042。我们进一步对改变数据集中的 SNP 位点数和人数来测试 EDCF 的假阳性率，测试结果（图 4.2(b)和(c)）显示 SNP 位点数和人数对 EDCF 假阳性率大小没有明显影响。当人数改变时，BOOST 的假阳性率变化不大（图 4.2(b)），但当 SNP 位点变大时，由于采用一个固定的阈值，BOOST 的假阳性率显著增长（图 4.2(c)）。

4.7.2 2-位点疾病模型

对于表 4.4 中 4 个有单位点边缘效应的 2-位点交互疾病模型，我们采用同 Wan *et*

al., 2010c 相同的参数，即模型 1 的 $h^2 = 0.03$，其他三个模型的 $h^2 = 0.02$，两个疾病相关的 SNP 位点的少数值频率相同，在测试中取三组值即 $f_m = 0.1, 0.2$ 或 0.4。我们首先在只含 100 个 SNP 位点的小数据集比较 EDCF 和 MB-MDR 的性能，测试结果（图 4.3）显示 EDCF 的性能随 α_s 的不同而发生变化，在大多数情况下，$\alpha_s = 0.05$ 时，EDCF 探测能力最强。当 $\alpha_s = 0.05$ 时，EDCF 对于模型 1 和 3 的探测能力强于 MB-MDR，特别是当 $f_m = 0.1$ 时，在其他情况下，EDCF 和 MB-MDR 有相似的探测能力。当 SNP 位点数较多时，MB-MDR 的运行相当缓慢，当位点数位 1000 时，MB-MDR 不能在我们可接受的时间内完成。

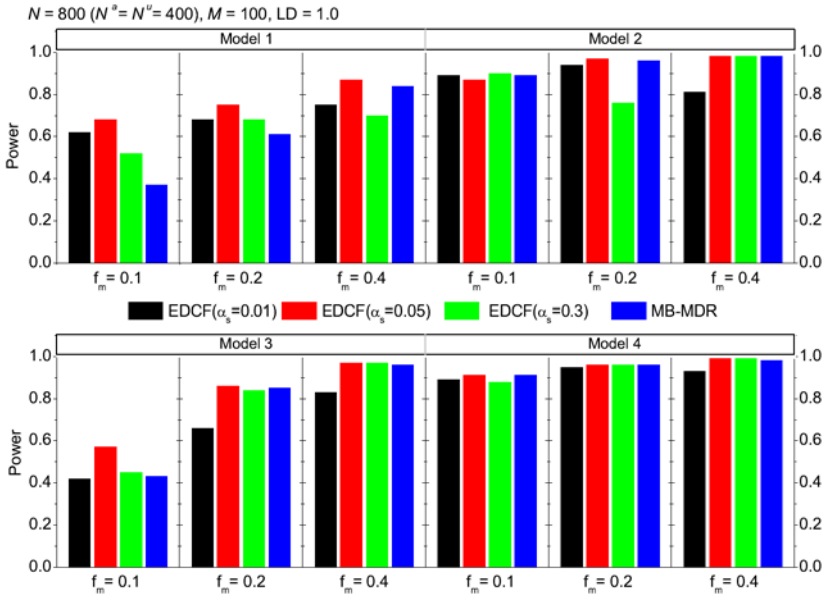


图 4.3 EDCF 和 MB-MDR 的性能比较。病例组和对照组的人数均为 400，连锁不平衡（LD）水平 $r^2 = 1$。

我们在包含 2000SNP 位点较大的数据集上对 EDCF、BOOST、SNPruler、epiMODE 和 ChiSQ 进行测试，测试结果如图 4.4 所示。当数据集中人数从 800 上升到 1600 和 LD 水平 r^2 从 0.7 变为 1.0 时，所有算法的探测能力都显著提高。对于模型 1 和 3，当植入的疾病相关位点的少数值频率 f_m 从 0.1 改变到 0.4 时，大多数算法的探测能力增强，而对于模型 2 和 4，这种趋势不明显。我们不清楚为何 BOOST 对模型 1 显示一种不同的趋势，虽然这种趋势同原始文献(Wan *et al.*, 2010c)相一致。当 $N = 800$，$r^2 = 0.7$ 时，所有算法性能都较差，可是除了少数情况（如对于模型 2 和 4，$f_m = 0.4$）和 BOOST 探测能力相差不多之外，其他情况下 EDCF 的探测能力是最高的。这些探测能力的差别大多具有统计显著性，其 *z*-统计量的 *p*-value 低于 0.01。例如，在所有

48 种中的 28 种情况下 EDCF 的探测能力统计显著的超过了 BOOST；而在其他 20 种情况下，EDCF 和 BOOST 的探测能力基本相同。在多种情况下，ChiSQ 的探测能力只比 EDCF 差一点，它通常比其他复杂的方法更有效。在模型 3 和 4 的某些情况下，BOOST 的探测能力优于 ChiSQ 。epiMODE 和 SNPRuler 的性能不稳定，在某些参数设置下，它们没有探测能力。

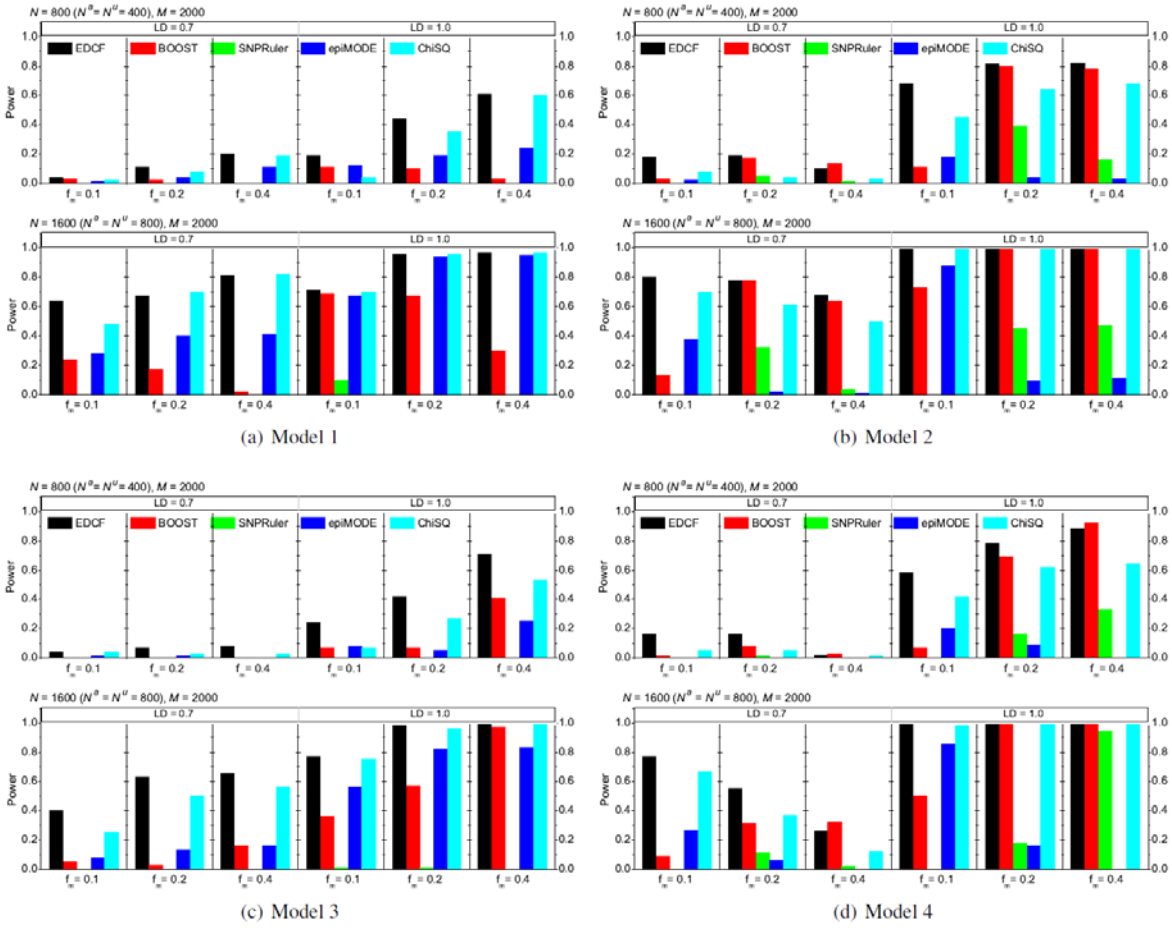


图 4.4 EDCF、BOOST、SNPRuler、epiMODE 和 ChiSQ 在模型 1、2、3 和 4 上性能比较。

另外在表 4.5 和 4.6 中的 60 个纯 2-位点交互模型（116）上我们也对算法进行了测试。这些模型也被 Wan 等[116]用于算法的测试比较，其遗传可能性 h^2 从 0.024 到 0.4，疾病相关位点的 f_m 为 0.2 或 0.4。对于每一个模型，模拟数据集中包含 800 个个体，其中病例组和对照组人数相等。实验结果如图 4.5 所示，当 $h^2 \geq 0.1$ 时，EDCF、BOOST、SNPRuler 和 ChiSQ 都有很强的探测能力，几乎达到 100%。可是当 $h^2 < 0.1$ 时，这些算法的探测能力明显下降。令人吃惊的是 epiMODE 对这 60 个疾病模型没有探测能力，这可能暗示它在探测纯 2-位点交互上有某些局限性。

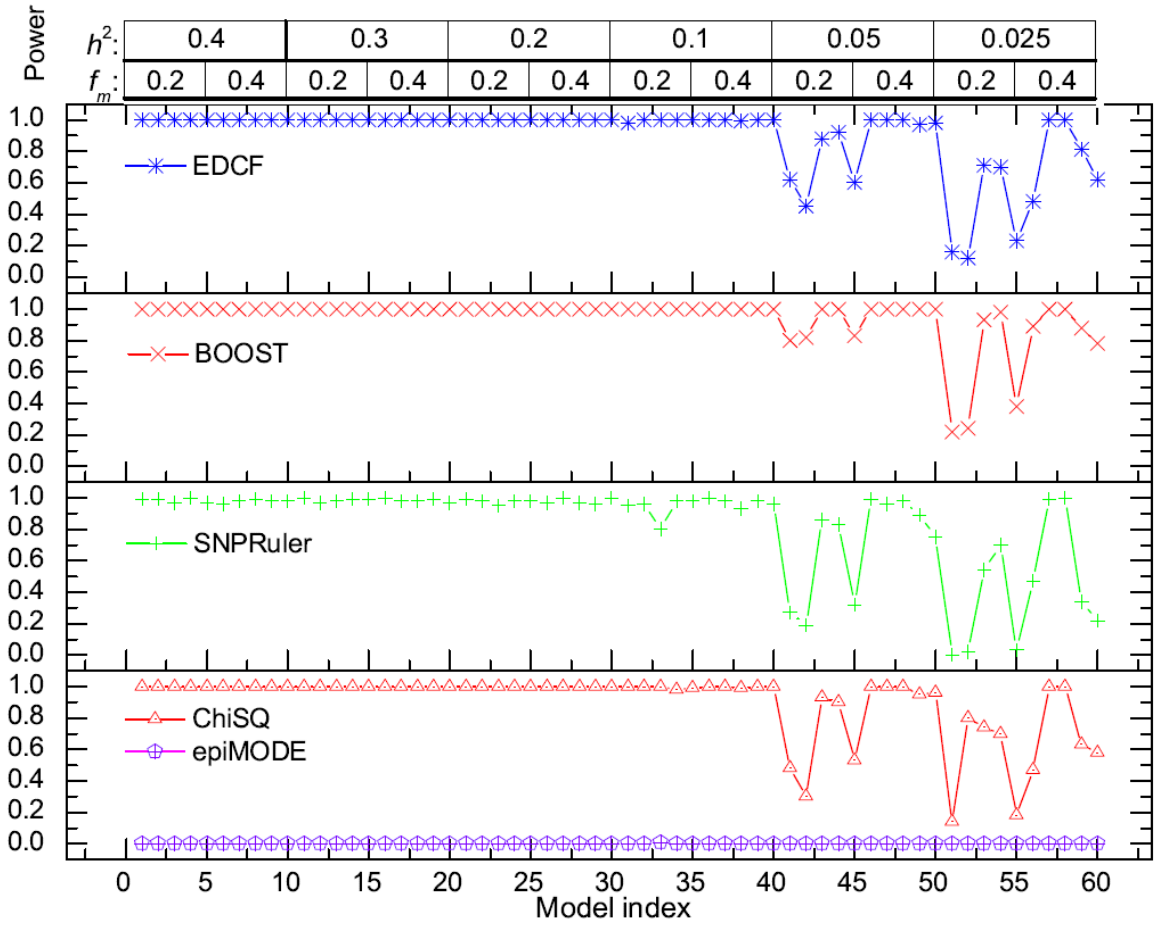
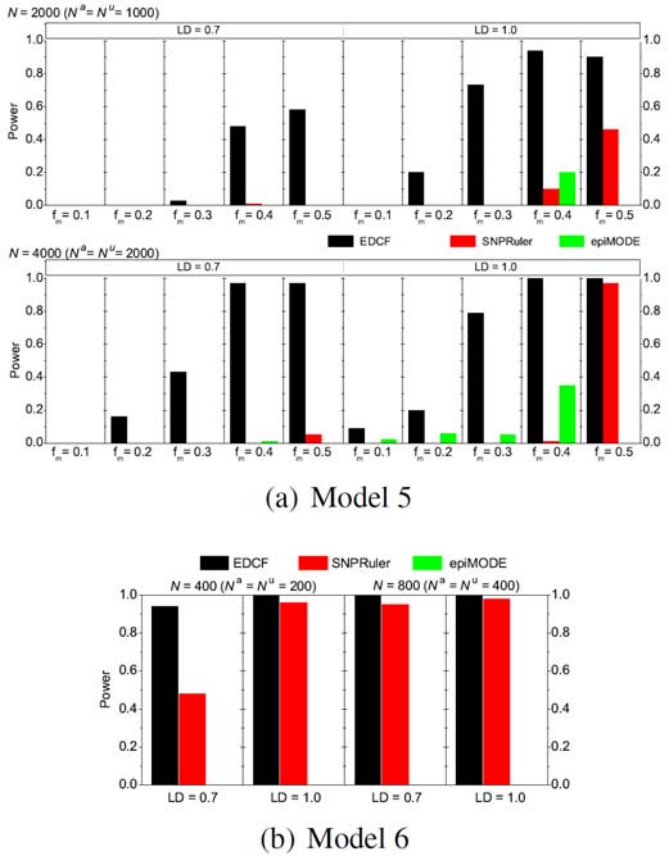


图 4.5 EDCF、BOOST、SNPRuler、epiMODE 和 ChiSQ 在 60 个没有单位点边缘效应的纯 2-位点交互模型（表 4.5 和 4.6）上性能比较。

4.7.3 3-位点疾病模型

在本节中，我们只对能明确输出 3-位点交互模块的算法进行测试，因为 BOOST 只能探测 2-位点的交互，所以本节不对它进行比较测试。使用 3-位点的 χ^2 统计量穷尽测试所有可能的 3-位点交互模块所需的计算资源过大，故本节也不对 ChiSQ 进行测试比较。因此在 3-位点疾病模型上，本节仅对 EDCF、SNPRuler 和 epiMODE 进行性能测试。模型 5 生成的数据集中人数 N 为 2000 或 4000，疾病相关位点的 f_m 从 0.1 到 0.5，当 $f_m = 0.1, 0.2, 0.3, 0.4$ 和 0.5 时，模型 5 疾病发病率奇异率列联表（表 4.7）中的参数 l 分别设置为 4, 1.5, 1, 0.7 和 0.5，其他参数 β 和 θ 通过数值计算方法确定以便使模型的单位点边缘效应 λ 固定为 0.2（详细情况请参照 118）。对于模型 6（表 4.7），数据集中人数 N 为 400 或 800，疾病相关位点的 f_m 固定为 0.5，人群的疾病发病率 $p = 0.01$。在模型 5 和 6 上的测试结果分别如图 4.6(a)和(b)所示。图 4.6(a)显示：

当 f_m 较小时，所有算法对模型 5 的探测能力都很弱，甚至没有探测能力，除了在 N = 4000，f_m = 0.5 且 LD 水平 r^2 = 1.0 的情况下，SNPRuler 的探测性能和 EDCF 相差不多，在其他情况下，EDCF 具有很强的探测能力，显著性地超过了 SNPRuler 和 epiMODE。图 4.6(b)显示：在模型 6 的各种参数设置中，EDCF 优于 SNPruler，而 epiMODE 对纯 3-位点交互模型没有探测能力。



(a) Model 5

(b) Model 6

图 4.6 2 个 3-位点疾病模型（表 4.7）上的性能比较。(a)有一些单位点边缘效应的模型 5。(b)没有边缘效应的模型 6。

4.7.4 运行时间

我们改变模拟数据集的人数 N 和 SNP 位点数 M 来比较算法的运行速度，实验结果如图 4.7 所示。当 M 固定不变时，除 epiMODE 的运行时间基本保持不变外其他所有算法的运行时间随着 N 的增大而线性增加（图 4.7(a)和(c)）。当 N 固定不变时，M 增大时，所有算法的运行时间成二次方增长（图 4.7(b)和(d)）。图 4.7 还显示在进行 2-位点交互模块探测时，EDCF、BOOST 和 ChiSQ 的运行时间没有明显的区别，相对来说 MB-MDR、SNPRuler 和 epiMODE 运行速度要慢很多。在进行 3-位点交互模块探测时，EDCF 比 SNPRuler 和 epiMODE 快许多。

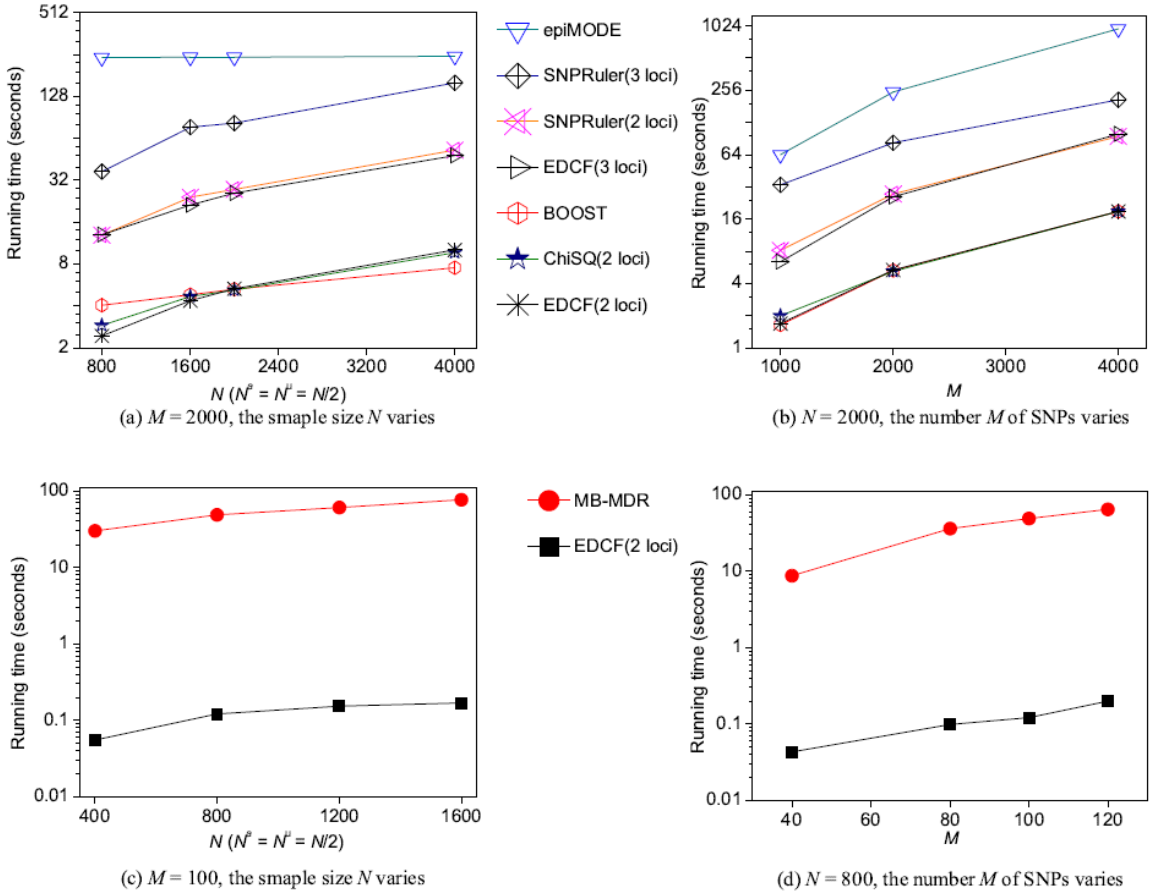


图 4.7 运行时间比较。(a) $M=2000$，数据集人数N从800增加到4000时EDCF、BOOST、SNPRuler、epiMODE和ChiSQ的运行时间比较。(b) $N=2000$，SNP位点数M从1000增加到4000时EDCF、BOOST、SNPRuler、epiMODE和ChiSQ的运行时间比较。(c) $M=100$，数据集人数N从400增加到1600时EDCF和MB-MDR运行时间比较。(d) $N=800$，SNP位点数M从40增加到120时EDCF和MB-MDR运行时间比较。

4.7.5 真实 GWAS 数据集上的测试

老年黄斑变性（Age-related macular degeneration, AMD）是50岁以上老年人致盲的首要原因，是一种常见的老年相关逐步损害视力的常见眼病。我们利用AMD数据集[126]测试EDCF的探测能力。AMD数据集包含96个患者和50个健康人在103,611 SNP位点上的基因型数据，去掉同构和信息不完整（有5个以上的人缺失基因型）的SNP位点后，剩下96,607个SNP位点。测试时EDCF的参数设置如下：$k=20$，进行单个位点、2个、3个和4个SNP位点交互分析时，α_0 分别采用 0.05、0.02、0.002 和0.00002，在进行3个和4个SNP位点交互分析时f_s 分别采用 2000和400。

对于AMD数据集，Klein等[126]基于1个自由度的SNP位点值关联测试探测出两个SNP位点（rs380390和rs1329428）与AMD有关联，EDCF利用基于基因型的2个自由度的的关联测试发现者两个位点的在单位点关联分析中排在前2位，它们未经修正的

p-vlaue分别是1.75×10^{-6}和4.61×10^{-6}，但是经Bonferroni校正后，它们没有达到我们预先规定的统计显著性阈值0.05，这跟Zhang和Liu[118]及Wan等[115]的结论一致。

EDCF没有发现有统计显著性的2-位点交互模块，但是发现了一些具有统计显著性的3-位点和4-位点交互模块。虽然在单位点分析时，rs380390没有达到统计显著性阈值，但是它出现在EDCF探测到的一个3-位点交互模块(rs380390, rs3781868, rs1036995)之中，其未经修正的*p*-vlaue = 8.0×10^{-18}。rs380390位于1号染色体长臂上的CFH基因上，该基因编码的蛋白质在补体激活（complement activation）调控过程中有重要作用。rs3781868在位于11q22-q23的NPAT基因之上，其编码的蛋白质在细胞周期的G1和S阶段是必须的。rs1036995在位于13q21的PCDH9基因上，其编码的钙粘蛋白神经元受体（cadherin-related neuronal receptor）可能参与了特定的神经元连接和信号传导活动。EDCF 还发现了其他具有统计显著性的3-位点和4-位点交互模块: (rs1458402, rs2207768, rs4901408), (rs1476623, rs6967345, rs1408120, rs10506115) 和 (rs595113, rs1569651, rs2031175, rs9300104) ，它们的未经修正的*p*-vlaue分别是8.8×10^{-18}，3.2×10^{-24}和4.9×10^{-24}。这些模块的某些基因型组合在AMD患者中比在健康人中出现得频繁得多，可能暗示着这些模块中的SNP相关基因相互作用导致AMD疾病的形成。上述模块的基因型组合聚类信息见表4.8、4.9、4.10和4.11。

表 4.8 AMD 数据集中 3-位点交互模块(rs380390, rs3781868, rs1036995)的列联表

	Cases	Controls	Total	Genotypes of rs3781868, rs380390 and rs1036995
G_0	60	2	62	001, 012, 021, 101, 102, 200, 202, 211, 212
G_1	30	11	41	002, 020, 100, 110, 111, 201, 210, 220
G_2	6	37	43	000, 010, 011, 022, 112, 120, 121, 122, 221, 222
Total	96	50	146	

表 4.9 AMD 数据集中 3-位点交互模块(rs1458402, rs2207768, rs4901408)的列联表

	Cases	Controls	Total	Genotypes of rs4901408, rs1458402 and rs2207768
G_0	48	1	49	001, 002, 012, 100, 111, 112, 201, 212, 221, 222
G_1	44	14	58	000, 010, 011, 021, 022, 102, 110, 121, 122, 210, 220
G_2	4	35	39	020, 101, 120, 200, 202, 211
Total	96	50	146	

表 4.10 AMD 数据集中 4-位点交互模块(rs1476623, rs6967345, rs1408120, rs10506115)的列联表

	Cases	Controls	Total	Genotypes of rs10506115, rs6967345, rs1408120 and rs1476623
G_0	62	0	62	0200, 0211, 1020-1022, 1102, 1110, 1120, 1122, 1211, 1220, 1221, 2000, 2001, 2011, 2022, 2100, 2220
G_1	31	7	38	others
G_2	3	43	46	0021, 1001, 1011, 1101, 1102, 2012, 2020, 2110-2122
Total	96	50	146	

表 4.11 AMD 数据集中 4-位点交互模块(rs595113, rs1569651, rs2031175, rs9300104)的列联表

	Cases	Controls	Total	Genotypes of rs2031175, rs1569651, rs595113 and rs9300104
G_0	61	0	61	0012, 0212, 0221, 1001, 1012, 1021, 1110, 1112, 1121, 1212, 2021, 2110, 2120-2122, 2210, 2221, 2222
G_1	34	10	44	others
G_2	1	40	41	0110, 0111, 0122, 1011, 1022, 1101, 1102, 1120, 1122, 1211, 2011, 2112, 2220
Total	96	50	146	

4.8 本章小结

基于相对频繁项聚类我们设计了一个新算法 EDCF，EDCF 把所有 *d*-位点的基因型组合划分为 3 组，然后用 χ^2 进行统计显著性评估。我们结合排列测试（pemutation test）和 Bonferroni 纠正的优点提出了一种快速的多重测试修正方法控制 EDCF 的假阳性率。大量模拟测试显示 EDCF 比最近提出的探测 GWAS 多位点交互模块的一些算法具有更强的探测能力。在运行时间上，EDCF 和 BOOST 进行 2-位点探测时具有相同的运行速度，比 MB-MDR、SNPRuler 和 epiMODE 快很多。在一个真实的 GWAS 数据集 AMD 上 EDCF 发现了一些基因型组合在患者中特别频繁，相关的基因交互模块可能与 AMD 相关联。

5 结论及展望

5.1 结论

SNP 是一个物种中不同个体表型的主要遗传来源，SNP 遗传标签在遗传疾病诊断、药物研究、个体识别等生物医学各方面有广泛的应用。本文对与 SNP 应用的三个重要的生物信息学问题单体型组装、人类白细胞抗原(Human Leukocyte Antigen, HLA)等位基因推断和全基因组关联分析 (Genome-wide association studies, GWAS)进行了深入研究。

（1）单体型组装

由于测序技术的制约，DNA 测序实验中能直接测定的片段所覆盖的最大 SNP 位点数 k 通常比较小的事实（k 通常小于 10），基于以上事实，本文对 MEC/GI 设计了一个时间复杂度为 $O(mk2^k+mlogm+mk)$、空间复杂度为 $O(mk2^k)$新的参数化精确算法，其中 m 为 SNP 矩阵的行数。实验结果表明， K-MEC/GI 和 Wang 等[63]的对应的分支限界算法具有相同的重构精度，而在片段数 m 达到 100，Wang 等提出的分支限界算法已无法运行的情况下，K-MEC/GI 和 Wang 等提出的遗传算法一样，仍然能快速运行。作为精确算法，K-MEC/GI 在单体型重构精度上比 Wang 等[63]对应的遗传算法有明显优势。考虑到生物问题的最优解往往不是唯一的，快速的能提供最优多个最优解的算法更能满足生物学家的需求，本文对进而设计了一个能提供 top-k 个 MEC 模型优化解的遗传算法 k-MD，大量实验表明 k-MD 在单体型重建率和运行速度上要明显好于 Wang 等[63]对应的求解 MEC 问题的遗传算法 GA-MEC。

（2） 人类白细胞抗原(Human Leukocyte Antigen, HLA)等位基因推断

HLA 基因的不同变异与许多免疫疾病、炎症和感染有关联，可是用生物实验测试方法直接确定 HLA 基因变异耗时耗力，制约了有关 HLA 基因的大规模研究。本文基于单体型相似加权图设计了一个 HLA 基因推断算法 WSG-HI，该算法根据一个群体的 SNP 基因型数据和其中多数个体的 HLA 基因信息推断出其余个体的 HLA 基因。在一组预先测定了 SNP 基因型和 HLA 基因的数据集上的大量实验测试显示 WSG-HI

能根据 HLA 基因相邻区域的 SNP 基因型数据精确地进行 HLA 基因推断。和最近提出的一个基于 IBD 的算法[2]相比，WSG-HI 在推断 HLA-A 和 HLA-B 时达到了相同的精度，在推断 HLA-C、HLA-DRB1、HLA-DQA1 和 HLA-DQB 时，WSG-HI 更精确。

（3）全基因组关联分析 (Genome-wide association studies, GWAS)

目前 GWAS 采用的主要模式是疾病与单个 SNP 位点相关统计分析的方法，可是人类复杂疾病往往是多个基因的交互作用(epistasis)的结果。本文提出了一个基于相对频繁项的多位点交互探测算法（Epistasis Detector based on the Clustering of relatively Frequent items，EDCF）。EDCF 采用穷尽搜索方法搜索两位点的交互模块，然后在此基础上采用步进的方法进行高阶交互的寻找。EDCF 把所有的基因型组合聚类成三组，分别代表发病组频繁基因型组合、控制组频繁基因型组合和其他基因型组合，聚类的统计显著性的大小用皮尔逊卡方统计（Pearson χ^2）来评估。大量模拟测试结果显示 EDCF 比最近提出的 MB-MDR[119]、BOOST[117]、SNPRuler[116]和 epiMODE[114]能更快更强地发现多位点交互模块。

5.2 展望

作为一门结合生物学、计算机科学、统计学等的新兴交叉学科，生物信息学（Bioinformatics）受到国外研究者的高度重视。随着人类基因组计划的提前完成，大量有关生物信息学的研究成果涌现在国际权威刊物和知名学术会议上。近年来，我国诸多学者也开始从事生物信息学的研究。对于生物信息学中出现的大量 NP-难问题，本文采用小参数方法、人工智能方法、图论方法、聚类和统计等方法在单体型型组装、HLA 推断和 GWAS 算法设计取得了成功的应用，这些算法在实际应用中具有较高的精度和较快的运行速度。当然，本文的工作还是存在一些不足，在相关的研究领域还存在着大量悬而未决的问题，下面结合本文的内容，提出一些未来可以深入研究的方向：

(1) 现有的单体型组装问题的各计算模型都只要求一个最优解，而生物学家可能对多个解更感兴趣，因此枚举出多个最优解或最近似的几个解是未来可以研究的方向。虽然本文提出的一个遗传算法能提供 MEC 模型的 top-k 个优化解，可是遗传算法不能确保这些解是最优的，目前缺乏求解 top-k 个最优解的精确算法。

由于单链结构，RNA 病毒复制变异率高，加上复制过程中可能发生重组事件，因此对于病毒而言，每个位点均可能发生变异，因此同一患者中的同一种病毒可以

具有多个不同的单体型，利用超高深度测序技术获得病毒群 RNA 片段数据重构多个不同的单体型的优化算法将是我们下一步要研究的内容。

(2) 本文提出的 HLA 推断算法是需要家族信息才能有较高的推断精度，这就局限了它的应用范围，不需要家族信息的 HLA 推断算法更加具有适应性，如何集成大量无关个体的基因型和相关 HLA 基因信息，根据个体基因型精确推断 HLA 基因是一个挑战性的问题。

(3) 本文提出的多基因交互探测算法没有考虑人口层化（population stratification）问题，很小的人口层化将显著增加算法的假阳性率，如何集成一些处理人口层化的策略使算法更稳健值得进一步研究。

参考文献

[1] International Human Genome Sequencing Consortium. Finishing the euchromatic sequence of the human genome. Nature, 2004, 431:931 - 945.

[2] S. Levy, G. Sutton, P. C. Ng, et al. The Diploid Genome Sequence of an Individual Human. PLoS Biology, 2007, 5(10):e254.

[3] P. Taillon-Miller, Z. Gu, Q. Li, et al. Overlapping Genomic Sequences: A Treasure Trove of Single-Nucleotide Polymorphisms. Genome research, 1998, 8(7): 748-754.

[4] 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

[5] G.A. Thorisson, L. D. Stein. The SNP Consortium website: past, present and future. Nucleic Acids Research, 2003, 31(1):124-127.

[6] The International HapMap Consortium. A haplotype map of the human genome. Nature, 2005, 437(7063):1299-1320.

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

[8] The International HapMap Consortium. A second generation human haplotype map of over 3.1 million SNPs. Nature, 2007, 449:851-861.

[9] G. Lancia, V. Bafna, S. Istrail, et al. SNPs problems, complexity, and algorithms. In: F. M. Heide (ed.). Proc. Ann. European Symp. on Algorithms (ESA). Berlin -Heidelberg: Springer-Verlag, 2001. Lecture Notes in Computer Science, 2161: 182-193.

[10] D. Gusfield. Inference of haplotypes from samples of diploid populations: complexity and algorithms. J. Computational Biology, 2001, 8(3):305-324.

[11] T. Shiina, K. Hosomichi, H. Inoko, J.K. Kulski. The HLA genomic loci map: expression, interaction, diversity and disease. Journal of Human Genetics, 2009, 54:15-39.

[12] M. N. Setty, A. Gusev, I. Pe'er. HLA Type Inference via Haplotypes Identical by Descent. In Proceedings of 14th Annual International Conference of Research in Computational Molecular Biology (RECOMB 2010): 12-15 August 2010; Lisbon, Portugal, Volume 6044 of LNBI. Edited by Berger B, Berlin Heidelberg: Springer-Verlag; 2010: 491-505.

[13] C. Vandiedonck, J.C. Knight. The human Major Histocompatibility Complex as a

paradigm in genomics research. Briefings in Functional Genomics and Proteomics, 2009, 8(5):6.

[14] P. I. W. de Bakker, G. Mcvean, P. C. Sabeti, et al. A high-resolution HLA and SNP haplotype map for disease association studies in the extended human MHC. Nature Genetics, 2006, 38(10):1166-1172.

[15] L. Handunnetthi, S. V. Ramagopalan, G. C. Ebers, J. C. Knight. Regulation of major histocompatibility complex class II gene expression, genetic variation and disease. Genes Immun, 2010, 11(2):99-112.

[16] S. Leslie, P. Donnelly, G. McVean. A statistical method for predicting classical HLA alleles from SNP data. American Journal of Human Genetics, 2008, 82:48-56.

[17] Wellcome Trust Case Contorl Consotium. Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. Nature, 2007, 447: 661–78.

[18] P. E. Stuart, et al. Genome-wide association analysis identifies three psoriasis susceptibility loci. Nat. Genet., 2010, 42:1000–4.

[19] E. Melum, et al. Genome-wide association analysis in primary sclerosing cholangitis identifies two non-hla susceptibility loci. Nat. Genet., 2011, 43: 17–9.

[20] D. B. Goldstein. Common genetic variation and human traits. N. Engl. J. Med., 2009, 360:1696–8.

[21] J. Hardy, A. Singleton. Genomewide association studies and human disease. N. Engl. J. Med., 2009), 360: 1759–68.

[22] Q. He, D. Y. Lin. A variable selection method for genome-wide association studies. Bioinformatics, 2011, 27: 1–8.

[23] H. J. Cordell. Detecting gene-gene interactions that underlie human diseases. Nat. Rev. Genet., 2009, 10: 392–404.

[24] P. C. Phillips. Epistasis–the essential role of gene interactions in the structure and evolution of genetic systems. Nat. Rev. Genet., 2008, 9: 855–67.

[25] M. D. Ritchie, *et al.* Multifactor-dimensionality reduction reveals high-order interactions among estrogen-metabolism genes in sporadic breast cancer. Am. J. Hum. Genet., 2001, 69, 138–47.

[26] M. R. Nelson, *et al.* A combinatorial partitioning method to identify multilocus genotypic partitions that predict quantitative trait variation. Genome Res., 2001, 11, 458–470.

[27] 李婧, 潘玉春, 李亦学, et al. 人类基因组单核苷酸多态性和单体型的分析及应用. 遗传学报, 2005, 32(8):879-889.

[28] A. G. Clark. The role of haplotypes in candidate gene studies. Genet Epidemiol, 2004,

27(4):321-33.

[29] I. Tachmazidou, C. J. Verzilli, M. D. Iorio. Genetic association mapping via evolution-based clustering of haplotypes. PLoS Genet, 2007, 3(7):e111.

[30] B. L. Browning, S. R. Browning. Efficient multilocus association testing for whole genome association studies using localized haplotype clustering. Genet Epidemiol, 2007, 31(5):365-75.

[31] R. W. Morris, N. L. Kaplan. On the advantage of haplotype analysis in the presence of multiple disease susceptibility alleles. Genet Epidemiol, 2002, 23(3): 221-33.

[32] M. P. Epstein, G. A. Satten. Inference on haplotype effects in case-control studies using unphased genotype data. Am J Hum Genet, 2003, 73(6):1316-29.

[33] M. Zoledziewska, C. Perra, V. Orru, et al. Further evidence of a primary, causal association of the PTPN22 620W variant with type 1 diabetes. Diabetes, 2008, 57(1):229-34.

[34] H. Zhou, L. J. Wei, X. Xu, et al. Combining association tests across multiple genetic markers in case-control studies. Hum Hered, 2008, 65(3):166-74.

[35] S. Adamovic, S. S. Amundsen, B. A. Lie, et al. Fine mapping study in Scandinavian families suggests association between coeliac disease and haplotypes in chromosome region 5q32. Tissue Antigens, 2008, 71(1):27-34.

[40] M. van Oijen, E. Y. Cheung, C. E. Geluk, et al. Haplotypes of the fibrinogen gene and cerebral small vessel disease. The Rotterdam scan study. J Neurol Neurosurg Psychiatry, 2007.

[41] 彭子文, 陈晓岗, 唐劲松, 等. 生物钟基因隐花色素-1 与精神分裂症核心家系的单体型相对风险分析. 中国神经精神疾病杂志, 2007, 33(8): 476-477.

[42] 刘敏, 凌四海, 李文标, 等. BDNF、GRIN1 基因与双相情感障碍的关联研究. 遗传, 2007, 29(1): 41-46.

[43] 金明娟, 陈坤, 张爽爽, 等. XRCC1 单核苷酸多态及单体型分布与乳腺癌的相关研究. 浙江大学学报：医学版, 2006, 35(4):370-376.

[44] 陈汉奎, 冯炳健, 梁慧, 等. 单体型分析将家族性鼻咽癌易感基因定位于 4p11～p14 区域. 科学通报, 2003, 48(16):1776-1779.

[45] D. Botstein, N. Risch. Discovering genotypes underlying human phenotypes: past successes for mendelian disease, future approaches for complex disease. Nat Genet, 2003, 33 Suppl:228-37.

[46] A. G. Clark. Inference of haplotypes from PCR-amplified samples of diploid populations. Mol. Biol. Evol., 1990, 7(2):111-122.

[47] D. Gusfield. A practical algorithm for optimal inference of haplotypes from diploid

populations. In: Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB 2000), 2000. 183-189.

[48] D. Gusfield. Haplotyping as perfect phylogeny: Conceptual framework and efficient solutions. In: Proceedings of the Annual International Conference on Computational Molecular Biology (RECOMB). Washington, DC, United States: Association for Computing Machinery, 2002. 166-175.

[49] D. Gusfield, S. Eddhu, C. Langley. Efficient reconstruction of phylogenetic networks with constrained recombination. In: Proceedings of the 2003 IEEE CSB Bioinformatics Conference, 2003. 363-374.

[50] D. Gusfield. Combinatorial approaches to haplotype inference. Lecture Notes in Computer Science. 2004. 2983: 136-136.

[51] D. Gusfield, S. Eddhu, C. Langley. Optimal, efficient reconstruction of phylogenetic networks with constrained recombination. Journal of Bioinformatics and Computational Biology, 2004, 2(1):173-213

[52] D. Gusfield. An overview of combinatorial methods for Haplotype Inference. In: S. Istrail, et. al. (Eds.). Proceedings of Computational Methods for SNPs and Haplotype Inference. Berlin Heidelberg: Springer-Verlag, 2004. LNBI, 2983: 9-25.

[53] G. Lancia. Integer programming models for computational biology problems. Journal of Computer Science and Technology, 2004, 19(1):60-77.

[54] G. Lancia, R. Rizzi. A polynomial case of the parsimony haplotyping problem. Operations Research Letters, 2006, 34(3):289-295.

[55] V. Bafna, B. V. Halldorsson, R. Schwartz, et al. Haplotypes and informative SNP selection algorithms: don't block out information. In: Proceedings of the Annual International Conference on Computational Molecular Biology (RECOMB). Washington, DC, United States: Association for Computing Machinery, 2003.19-27.

[56] V. Bafna, D. Gusfield, G. Lancia, et al. Haplotyping as Perfect Phylogeny: A direct approach. Journal of Computational Biology, 2003, 10(3-4):323-340.

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

[58] R. Rizzi, V. Bafna, S. Istrail, et al. Practical algorithms and fixed-parameter tractability for the single individual SNP haplotyping problem. Lecture Notes in Computer Science, 2002. 2452: 29-43.

[59] J. Li, T. Jiang. Efficient rule-based haplotyping algorithms for pedigree data. In: Proceedings of the Annual International Conference on Computational Molecular Biology (RECOMB), Washington, DC, United States: Association for Computing Machinery, 2003. 197-206.

[60] J. Li, T. Jiang. Efficient inference of haplotypes from genotypes on a pedigree. Journal of Bioinformatics and Computational Biology (JBCB), 2003, 1(1):41-69.

[61] J. Li, T. Jiang. An exact solution for finding minimum recombinant haplotype configurations on pedigrees with missing data by integer linear programming. In: Proceedings of the Annual International Conference on Computational Molecular Biology (RECOMB). Washington, DC, United States: Association for Computing Machinery, 2004. 20-29.

[62] L. Wang, Y. Xu. Haplotype inference by maximum parsimony. Bioinformatics, 2003, 19(14):1773-80.

[63] R. Wang, L. Wu, Z. Li, et al. Haplotype reconstruction from SNP fragments by minimum error correction. Bioinformatics, 2005, 21(10):2456-62.

[64] Y. Zhao, L. Wu, J. Zhang, et al. Haplotype assembly from aligned weighted SNP fragments. Computational Biology and Chemistry, 2005, 29(4):281-287.

[65] X. Zhang, R. Wang, L. Wu, et al. Models and Algorithms for Haplotyping Problem. Current Bioinformatics, 2006, 1(1):105-114.

[66] Z. Li, W. Zhou, X. Zhang, et al. A parsimonious tree-grow method for haplotype inference. Bioinformatics, 2005, 21(17):3475-81.

[67] X. Zhang, R. Wang, L. Wu, et al. Minimum conflict individual haplotyping from SNP fragments and related genotype. Evolutionary Bioinformatics, 2006, 2:271-280.

[68] 王瑞省, 吴凌云, 张继红, 等. 单体型装配问题及其算法. 高校应用数学学报: A 辑 2004, 19(B12):515-528.

[69] 李珍萍, 王勇, 赵玉英, 等. 单体型推断问题与配对图. 高校应用数学学报: A 辑, 2004, 19(B12):567-576.

[70] F. Y. L. Chin, Q. Zhang, H. Shen. k-recombination haplotype inference in pedigrees. Lecture Notes in Computer Science, 2005, 3515: 985-993.

[71] Q. Zhang, F. Chin, H. Shen. Minimum Parent-Offspring Recombination Haplotype Inference in Pedigrees. Lecture Notes in Computer Science, 2005, 3680:100-112.

[72] Q. Zhang, Y. Xu, G. Chen, et al. Maximum-Likelihood Estimation of Haplotype Frequencies in trio pedigrees. In: Proceedings of the First International Multi-Symposiums on Computer and Computational Sciences (IMSCCS'06), 2006. 35-39.

[73] 张强锋, 车皓阳, 陈国良, 等. 最大节约原则下单倍型推导问题的实用算法(英文). 软件学报, 2005, 16(10):1699-1707.

[74] 张强锋, 徐云, 陈国良, 等. 三元家庭基因数据的单体分型和单体型频率估计(英文). 软件学报, 2007, 18(09):2090-2099.

[75] 张强锋, 陈国良, 孙广中. 最大节约原则下单体型推导问题的复杂性. 中国科学

技术大学学报, 2006, 36(2):213-218.

[76] M. Xie, J. Wang, et al. Computational Models and Algorithms for the Single Individual Haplotyping Problem. Current Bioinformatics, 2010, 5(1): 18-28.

[77] 谢民主, 陈建二, 王建新. 个体单体型问题参数化算法研究. 计算机学报, 2009, 32(8): 1637-1650.

[78] M. Xie, J. Wang, J. Chen. A model of higher accuracy for the individual haplotyping problem based on weighted SNP fragments and genotype with errors. Bioinformatics, 2008, 24(13): i105-13.

[79] M. Xie, J. Wang. An Improved (and Practical) Parameterized Algorithm for the Individual Haplotyping Problem MFR with Mate-Pairs. Algorithmica, 2008, 52(2): 250-266.

[80] 谢民主, 陈建二, 王建新. 有Mate-Pairs的个体单体型MSR问题的参数化算法. 软件学报, 2007, 18(9): 2070-2082.

[81] R. Lippert, R. Schwartz, G. Lancia, et al. Algorithmic strategies for the single nucleotide polymorphism haplotype assembly problem. Briefings in Bioinformatics, 2002, 3(1):1-9.

[82] W. L. Hsu. A simple test for the consecutive ones property. Journal of Algorithms, 2002, 43(1):1-16.

[83] K. S. Booth, G. S. Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. Journal of Computer and System Sciences, 1976, 13(3):335-379.

[84] R. Cilibrasi, L. Iersel, S. Kelk, et al. On the complexity of several haplotyping problems. Lecture Notes in Computer Science, 2005, 3692: 128-139.

[85] R. Cilibrasi, L. van Iersel, S. Kelk, et al. The complexity of the single individual SNP haplotyping problem. Algorithmica, 2007, 49(1):13-36.

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

[87] W. Qian, Y. Yang, N. Yang, et al. Particle swarm optimization for SNP haplotype reconstruction problem. Applied Mathematics and Computation, 2008, 196(1):266-272.

[88] H. Greenberg, W. Hart, G. Lancia. Opportunities for combinatorial optimization in computational biology. INFORMS Journal on Computing, 2004, 16(3): 211-231.

[89] Y. Wang, E. Feng, R. Wang, et al. The haplotype assembly model with genotype information and iterative local-exhaustive search algorithm. Computational Biology and Chemistry, 2007, 31(4):288-293.

[90] L. Genovese, F. Geraci, M. Pellegrini. A fast and accurate heuristic for the single individual SNP haplotyping problem with many gaps, high reading error rate and low coverage. In: R. Giancarlo,S. Hannenhalli (eds.). Proc. WABI 2007. Berlin Heidelberg: Springer-Verlag, 2007. LNCS, 4645: 49-60.

[91] L. Li, J. H. Kim, M. S. Waterman. Haplotype reconstruction from SNP alignment. In: Proc. the Annual International Conference on Computational Molecular Biology (RECOMB 2006). Washington, DC, United States: Association for Computing Machinery, 2003. 207-216.

[92] S. Wernicke. On the algorithmic tractability of single mucleotide polymorphism (SNP) analysis and related problems [Ph. D.]. Universität Tűbingen, 2003.

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

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

[95] J. C. Venter, M. D. Adams, E. W. Myers, et al. The sequence of the human genome. Science, 2001, 291(5507):1304-1351.

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

[97] M. J. Daly, J. D. Rioux, S. F. Schaffner, et al. High-resolution haplotype structure in the human genome. Nature Genetics, 2001, 29(2):229-232.

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

[99] F. Hüffner. Algorithm engineering for optimal graph bipartization. Lecture Notes in Computer Science, 2005, 3503:240-252.

[100] W. Ansorge, H. Voss, U. Wirkner, 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.

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

[101] G. Myers. A dataset generator for whole genome shotgun sequencing. In: T. Lengauer, R. Schneider, P. Bork, et al. (Eds.). Proc. 7th Int'l Conf. Intelligent Systems for Molecular Biology (ISMB 99). California: AAAI Press, 1999, 202-210.

[102] P. Bonizzoni, G. Della Vedova, R. Dondi, et al. The haplotyping problem: an overview of computational models and solutions. Journal of Computer Science and Technology, 2003, 18(6):675-688.

[103] J. M. Barker, T. M. Triolo, T. A. Aly, et al. Two single nucleotide polymorphisms identify the highest-risk diabetes HLA genotype: potential for rapid screening. Diabetes, 2008, 57(11):3152-3155.

[104] A. J. Monsuur, P. I. de Bakker, A. Zhernakova, et al. Effective detection of human leukocyte antigen risk alleles in celiac disease using tag single nucleotide polymorphisms. PLoS One, 2008, 3(5):e2270.

[105] L. Koskinen, J. Romanos, K. Kaukinen, et. al. Cost-effective HLA typing with tagging SNPs predicts celiac disease risk haplotypes in the Finnish, Hungarian, and Italian populations. Immunogenetics 2009, 61(4):247-256.

[106] A. Gusev, J. K. Lowe, M. Stoffel, et. al. Whole population, genome-wide mapping of hidden relatedness. Genome Research, 2009, 19(2):318-326.

[107] J Li, T Jiang. A survey on haplotyping algorithms for tightly linked markers. J Bioinform Comput Biol, 2008, 6:241-59.

[108] V Bansal, V Bafna. HapCUT: an efficient and accurate algorithm for the haplotype assembly problem. Bioinformatics, 2008, 24(16):i153-9.

[109] J Xiao, T Lou, T Jiang. An efficient algorithm for haplotype inference on pedigrees with a small number of recombinants (extended abstract). In Proceedings of 17th Annual European Symposium (ESA 2009): 7-9 September 2009; Copenhagen, Denmark, Volume 5757 of LNCS. Edited by Fiat A, Sanders P, Berlin Heidelberg: Springer-Verlag; 2009:325-336.

[110] X. Li, J. Li. An almost linear time algorithm for a general haplotype solution on tree pedigrees with no recombination and its extensions. Journal of Bioinformatics and Computational Biology, 2009, 7(3):521-545.

[111] D. E. Goldberg. Genetic algorithms in search, optimization and machine learning. Reading, MA: Addision-Wesley, 1989.

[112] J. Li, et al. (2011) The Bayesian lasso for genome-wide association studies. Bioinformatics, 2011, 27, 516-23.

[113] J. H. Moore, F. W. Asselbergs, S. M. Williams. (2010) Bioinformatics challenges for genome-wide association studies. Bioinformatics, 2010, 26, 445–55.

[114] W.Tang, et al. Epistatic module detection for case-control studies: a Bayesian model with a Gibbs sampling strategy. PLoS Genet., 2009, 5, e1000464.

[115] X. Wan, et al. Detecting two-locus associations allowing for interactions in genome-wide association studies. Bioinformatics, 2010, 26, 2517–25.

[116] X. Wan, et al. Predictive rule inference for epistatic interaction detection in genome-wide association studies. Bioinformatics, 2010, 26, 30–7.

[117] X. Wan, et al. BOOST: a fast approach to detecting gene-gene interactions in genome-wide case-control studies. Am. J. Hum. Genet., 2010, 87, 325–40.

[118] Y. Zhang, J. S. Liu. Bayesian inference of epistatic interactions in case-control studies. Nat. Genet., 2007, 39, 1167–73.

[119] T. Cattaert, et al. Model-based multifactor dimensionality reduction for detecting epistasis in case-control data in the presence of noise. Ann. Hum. Genet., 2011, 75, 78–89.

[120] J. Gui, et al. A robust multifactor dimensionality reduction method for detecting gene-gene interactions with application to the genetic analysis of bladder cancer susceptibility. Ann. Hum. Genet., 2011, 75, 20–8.

[121] M. D. Ritchie, et al. Optimization of neural network architecture using genetic programming improves detection and modeling of gene-gene interactions in studies of human diseases. BMC Bioinformatics, 2003, 4, 28.

[122] D. F. Schwarz, I. R. Konig, A. Ziegler. On safari to random jungle: a fast implementation of random forests for high-dimensional data. Bioinformatics, 2010, 26, 1752–8.

[123] J. Marchini, P. Donnelly, L. R. Cardon. Genome-wide strategies for detecting multiple loci that influence complex diseases. Nat. Genet., 2005, 37, 413–7.

[124] J. Li. A novel strategy for detecting multiple loci in genome-wide association studies of complex diseases. Int. J. Bioinformatics Research and Applications, 2008, 4, 150-63.

[125] R. A. Fisher. On the interpretation of χ^2 from contingency tables, and the calculation of P. Journal of the Royal Statistical Society, 1922, 85, 87-94.

[126] R. J. Klein, et al. Complement factor H polymorphism in age-related macular degeneration. Science, 2005, 308, 385–9.

致　谢

本人在中南大学信息与工程学院博士后流动站三年的博士后工作中得到了许多老师、同学和亲友的关心和帮助，在此谨向指导、关心和支持我的老师、同学和亲朋好友们致以我最真挚的感谢!

感谢我的合作导师桂卫华教授、王建新教授和陈松乔教授，其中王建新教授还是指导我博士论文的导师，他们以其渊博的知识、敏锐的学术洞察力、勤奋刻苦的科研精神、平易近人的工作作风和一丝不苟、精益求精的科研态度对我言传身教，使我受益终生。他们在科研上对我的指导和启发令我受益终生，在生活上对我的呵护和关心令我全家衷心感谢。

感谢美国加州大学河滨分校姜涛教授提供给我去美国研究的机会，在美期间科研上的细致指导及生活上周到的关心。同时感谢我的博士生导师陈建二教授对我的得力推荐。感谢美国 Case Western Reserve University 的 Jing Li 副教授，与他的一些讨论及他的修改使我的论文增色不少。

感谢李敏、黄家玮、彭小清及其他师弟妹们为我所做的不少工作。感谢中南大学信息与工程学院的领导、老师给我的帮助。

感谢陈晓岗教授、胡志刚教授、喻祖国教授、陶永光教授和王伟平教授在百忙之中参加本人博士后出站报告会。

感谢我年迈的双亲，亲爱的妻子和我那可爱的儿子。

本报告中的研究得到了中国博士后一等科研基金和中南大学博士后一等科研基金资助，也获得了国家自然科学基金（No. 61070145）资助，有部分研究内容还获得了美国 NIH 科研项目(No. 2R01LM008991)的资助，在此一并表示感谢。

博士生期间发表的学术论文，专著

[1] 谢民主, 陈建二, 王建新. 有 Mate-Pairs 的个体单体型 MSR 问题的参数化算法. 软件学报, 2007, 18(9): 2070-2082.

[2] Minzhu Xie, Jianer Chen, Jianxin Wang. Research on Parameterized Algorithms of the Individual Haplotyping Problem. Journal of Bioinformatics and Computational Biology, 2007, 5(3): 795-816.

[3] Minzhu Xie, Jianxin Wang. An Improved (and Practical) Parameterized Algorithm of the Individual Haplotyping Problem MFR with Mate-pairs. Algorithmica, 2008, 52(2): 250-266.

[4] 谢民主, 王建新, 陈建二. 单体型组装问题 MEC/GI 模型的参数化算法. 高技术通讯, 2008, 18(4): 422-428.

[5] Minzhu Xie, Jianxin Wang, Jianer Chen. A Practical Parameterized Algorithm for the Individual Haplotyping Problem MLF. TAMC 2008, LNCS 4978, pp. 439–450.

[6] Minzhu Xie, Jianxin Wang, Jianer Chen. A Practical Exact Algorithm for the Individual Haplotyping Problem MEC. BMEI (1) 2008: 72-76.

[7] Minzhu Xie, Jianxin Wang, Jianer Chen. A Practical Exact Algorithm for the Individual Haplotyping Problem MEC/GI. COCOON 2008 : 342-351.

[8] Minzhu Xie, Jianxin Wang, Jianer Chen. A High Accurate Model of the Individual Haplotyping Problem Based on Weighted SNP Fragments and Genotype with Errors. ISMB2008, Bioinformatics, 2008, 24(13): p. i105-113.

[9] Minzhu Xie, Jianxin Wang, Wei Zhou, Jianer Chen. A Practical Parameterized Algorithm for Weighted Minimum Letter Flips Model of the Individual Haplotyping Problem. FAW 2008

博士后期间发表的学术论文，专著

[1] 谢民主, 陈建二, 王建新. 个体单体型问题参数化算法研究. 计算机学报, 2009. 32(8): p. 1637-1650.

[2] Jianxin Wang, Minzhu Xie, Jianer Chen. A Practical Exact Algorithm for the Individual Haplotyping Problem MEC/GI. Algorithmica, 2010, 56(3): 283-296.

[3] Minzhu Xie, Jianxin Wang, Jianer Chen. A practical parameterised algorithm for the individual haplotyping problem MLF. Mathematical Structures in Computer Science, 2010, 20(5): 851-863.

[4] Xie, M., J. Wang, J. Chen, et al., Computational Models and Algorithms for the Single Individual Haplotyping Problem. Current Bioinformatics, 2010. 5(1): p. 18-28.

[5] 谢民主, 刘新求. 枚举单体型组装问题多个最优解的遗传算法设计. 计算机工程与应用, 2010, 46(11): 7-9.

[6] Minzhu Xie, Jing Li, Tao Jiang. Accurate HLA type inference using a weighted similarity graph. GIW2010, BMC Bioinformatics, 2010, 11(Suppl 11):S10.

[7] Minzhu Xie, Jing Li, Tao Jiang. Jiang, Detecting genome-wide epistases based on the clustering of relatively frequent items. Bioinformatics, 2011. http://bioinformatics.oxfordjournals.org/content/early/2011/11/02/bioinformatics.btr603, doi: 10.1093/bioinformatics/btr603.

个人简历

谢民主, 男, 博士, 副教授, 1969 年 10 月出生于湖南涟源市白马镇, 1983 年 9 月初中毕业后以优异成绩考入娄底中等师范学校学习，1986 年 7 月毕业分配回家乡从事中学教育教学工作直至 2000 年 8 月，2000 年 9 月录取为中南大学信息科学与工程学院公费研究生，于 2003 年 4 月计算机应用专业研究生毕业获工学硕士学位，同年 7 月进入湖南师范大学物理与信息科学学院任教至今，期间 2004 年 9 月进入中南大学信息科学与工程学院攻读博士学位，2008 年 5 月获工学博士学位，2008 年 12 月——2011 年 12 月在中南大学信息科学与工程学院进行博士后研究。研究领域：生物信息学，计算机算法。

永久通信地址

地址:　湖南省长沙市湖南师范大学物理信息与科学学院

邮政编码: 410081

电子邮箱: xieminzhu@sina.com