参赛密码 _____

（由组委会填写）

“华为杯”第十三届全国研究生
数学建模竞赛

学　　校	上海交通大学
参赛队号	**10248299**
队员姓名	**1.胡子豪**
	2.纪德益
	3.周瑛

参赛密码 _____

（由组委会填写）

“华为杯”第十三届全国研究生数学建模竞赛

题 目 具有遗传性疾病和性状的遗传位点分析

摘 要：

本文针对具有遗传性疾病和性状的遗传位点分析问题，使用假设检验的方法，分别采用卡方检验、逻辑回归、SKAT 以及 metaCCA 方法建立数学模型，使用 MATLAB 和 R 语言及其工具包进行编程，在合理的假设下，确定了与遗传性疾病或相关性状有关联的位点和基因，并且对发现的致病位点及基因从理论上进行了统计分析及检验。

针对问题一，结合生物学意义，以等位基因在样本中出现频率为依据区分同一位点的两种等位基因，并且根据基因型进行 0,1,2 三个数值编码。

针对问题二，首先依据最小等位基因频率（MAF）控制以及 Hardy-Weinberg 平衡控制对题目所给数据进行 SNP 质量筛选，剔除 97 个不符合质量要求的位点；然后分别使用卡方检验方法和逻辑回归模型对剩余位点进行建模，最后通过显著性检验对位点与遗传疾病 A 进行显著性水平分析，找出显著的致病位点，并结合两种模型综合分析确定了致病位点的合理性。

针对问题三，分别采用逻辑回归模型和 SKAT 模型对由位点组合的基因进行建模，通过假设检验的方法，确定了致病基因，最后结合问题二的结果以及对两种模型分别对其自变量的独立性假设分析，说明 SKAT 模型的结果比逻辑回归的结果更可靠。

针对问题四，首先采用 metaCCA 算法，得到位点与性状之间的典型关联系数，随后通过统计检验的方法，确定了与相关性状整体相关联的位点。该模型

解出的最优位点 rs12746773 与其余位点显著性水平差异巨大，说明该位点与题目所给的 10 个性状具有很强的关联。

本文亮点在于：1）对题目所给数据进行合理预处理，筛选出部分质量不达标位点；2）对发现的致病位点或基因都采用多种模型进行统计分析与检验，并且从理论上分析对比了不同模型的合理性；3) 模型的扩展性和可移植性比较强。

关键词：位点（SNPs），卡方检验，逻辑回归, 典型关联分析（CCA）

目 录

1、问题重述

1.1、问题背景

人的遗传密码由人体中的 DNA 携带。人体的每条染色体携带一个 DNA 分子。DNA 是由分别带有 A,T,C,G 四种碱基的脱氧核苷酸链接组成的双螺旋长链分子，该长链分子共有约 30 亿个碱基对。基因则是 DNA 长链中有遗传效应的一些片段。在组成 DNA 的碱基对（或对应的脱氧核苷酸）中，有一些特定位置的单个核苷酸经常发生变异引起 DNA 的多态性，我们称之为位点（SNPs）。染色体、基因和位点的结构关系见图 1.1。

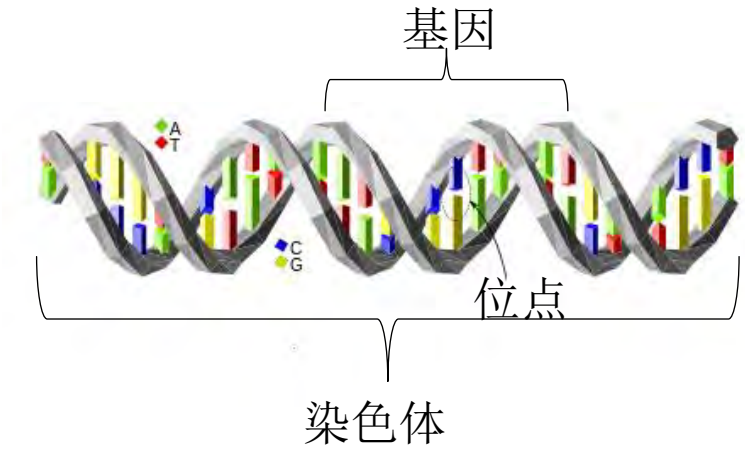

图 1.1 染色体、基因和位点的结构关系

在 DNA 长链中，位点个数约为碱基对个数的千分之一。由于位点在 DNA 长链中出现频繁，多态性丰富，它已成为人们研究 DNA 遗传信息的重要载体，被称为人类研究遗传学的第三类遗传标记。大量研究表明，人体的许多表型性状差异以及对药物和疾病的易感性等都可能与某些位点或和包含有多个位点的基因相关联。因此，定位与性状或疾病相关联的位点在染色体或基因中的位置，能帮助研究人员了解性状和一些疾病的遗传机理，也能使人们对致病位点加以干预，防止一些遗传病的发生。

近年来，研究人员大都采用全基因组（GWAS）的方法来确定致病位点或致病基因。具体做法是：招募大量志愿者（样本），包括具有某种遗传病的人和健康的人，通常用 1 表示病人，0 表示健康者。对每个样本，采用碱基(A,T,C,G)的编码方式来获取每个位点的信息(因为染色体具有双螺旋结构，所以用两个碱基的组合表示一个位点的信息)；如表 1 中，在位点 rs100015 位置，不同样本的编码都是 T 和 C 的组合，有三种不同编码方式 *TT*,*TC* 和 *CC*。类似地其他的位点虽然碱基的组合不同，但也只有三种不同编码。研究人员可以通过对样本的健康状况和位点编码的对比分析来确定致病位点，从而发现遗传病或性状的遗传机理。

表 1.1 在对每个样本采集完全基因组信息后，一般有以下的数据信息
(以 6 个样本为例，其中 3 个病人，3 个健康者)：

样本编号	样本健康状况	染色体片段位点名称和位点等位基因信息			
		rs100015	rs56341	...	rs21132
1	1	TT	CA	...	GT
2	0	TT	CC	...	GG
3	1	TC	CC	...	GG
4	1	TC	CA	...	GG
5	0	CC	CC	...	GG
6	0	TT	CC	...	GG

注：位点名称通常以 rs 开头。

1.2、问题提出

本题目针对某种遗传疾病(简称疾病 A)提供了 1000 个样本的信息，这些信息包括这 1000 个样本的疾病信息、样本的 9445 个位点编码信息，以及包含这些位点的基因信息。另外，人体的许多遗传疾病和性状是有关联的，科研人员往往把相关的性状或疾病放在一起研究以提高发现致病位点或基因的能力。本题也提供了上述 1000 个样本的 10 种相关性状的信息。

需要通过建立数学模型，在给定的数据下，求解以下 4 个问题：

问题一：请用适当的方法，把 genotype.dat 中每个位点的碱基(A,T,C,G) 编码方式转化成数值编码方式，便于进行数据分析。

问题二：根据附录中 1000 个样本在某条有可能致病的染色体片段上的 9445 个位点的编码信息(见 genotype.dat)和样本患有遗传疾病 A 的信息（见 phenotype.txt 文件）。设计或采用一个方法，找出某种疾病最有可能的一个或几个致病位点，并给出相关的理论依据。

问题三：同上题中的样本患有遗传疾病 A 的信息（phenotype.txt 文件）。现有 300 个基因，每个基因所包含的位点名称见文件夹 gene_info 中的 300 个 dat 文件，每个 dat 文件列出了对应基因所包含的位点(位点信息见文件 genotype.dat)。由于可以把基因理解为若干个位点组成的集合，遗传疾病与基因的关联性可以由基因中包含的位点的全集或其子集合表现出来。请找出与疾病最有可能相关的一个或几个基因，并说明理由。

问题四：在问题二中，已知 9445 个位点，其编码信息见 genotype.dat 文件。在实际的研究中，科研人员往往把相关的性状或疾病看成一个整体，然后来探寻与它们相关的位点或基因。试根据 multi_phenos.txt 文件给出的 1000 个样本的 10 个相关联性状的信息及其 9445 个位点的编码信息(见 genotype.dat)，找出与 multi_phenos.txt 中 10 个性状有关联的位点。

2、模型假设

- 假设样本的数据真实，来源可靠，且样本的患病及 SNP 信息正确；
- 假设样本具有普遍性且样本数量足够，可以用来反映性状表现型与位点或

位点组合的基因的关系；

- 假设每个 SNP 位点在所有样本中的分型成功率均在 75%以上，即符合 SNP 分型成功比例控制，可以通过该质量检测标准；
- 假设每个 SNP 位点均可通过孟德尔错误控制检测标准。

3、符号说明

A	主等位基因（major allele）
X	基因型信息（genotype）
Y	表现型信息（phenotype）
MAF	最小等位基因频率
χ^2	卡方统计量
dif	卡方统计量的自由度
$p\text{-}value$	假设检验中的显著性水平（P 值）
β	回归模型的回归系数
SE_{β_1}	回归系数 β_1 的标准差

4、问题一模型建立与求解

4.1、问题描述及分析

本题目的在于找寻合适的数值编码方式，便于后续对样本数据进行统计分析。

对于每一个位点，以位点 rs100015 位置为例，不同样本的编码都是 T 和 C 的组合，因此只对应三种编码方式 TT,TC 和 CC，其余位点只是参与组合编码的碱基不同，也是 3 种编码方式。因此，我们需要将这三种编码方式分别对应三个不同的数值。

解决本问题需要考虑的问题为：每一个位点都是由两种等位基因编码组成，在进行检验之前，这两种等位基因的地位是相同的，我们需要以何种标准来区分每一个位点的两种等位基因。

4.2、模型建立

设某 SNP 位点分别由两个等位基因编码，记为 A 和 a，在进行检验之前，

这两个等位基因的地位相同，为了区分两种等位基因，我们以 A 表示该位点所有样本中出现频率较大的等位基因（major allele）。则每一个 SNP 位点信息由 A 和 a（a 也被称为 minor allele）两种等位基因编码，形成了三个不同的基因型 AA,Aa(aA)和 aa。

因此对于每一个特定位点，我们根据基因型不同有如下编码：

$$U=\begin{cases}0 & if\ \ genotype\ is\ AA\\ 1 & if\ \ genotype\ is\ Aa\ or\ aA\\ 2 & if\ \ genotype\ is\ aa\end{cases} \qquad (4.1)$$

4.3、问题求解及分析

题目共给出 9445 个 SNP 位点，对于其中的每一个 SNP，我们分别统计该位点出现的两种等位基因在 1000 个样本中出现的频数，由于在杂合基因型中两种等位基因出现的频数相同，所以只需要比较该位点的纯合样本，比较得出频数较大的纯合样本，其对应的等位基因记为 A，另一种记为 a。然后再按照上述编码规则对该位点的样本进行数值编码。

在解决本问题时，我们以每一个位点的等位基因在所有样本中的出现频率作为标准来区分不同等位基因，以此进行编码。这种方式是具有生物学上的理论依据的。在生物统计学中，最小等位基因频率（minor allele frequency,MAF）有着很重要的统计意义，它会影响统计性能。最小等位基因指的就是该位点等位基因中出现频率较小的等位基因的出现频率。因此，我们在编码的时候就对其进行区分。

5、问题二模型建立与求解

5.1、问题描述及分析

本题的目的是设计一种方法，找出与某遗传疾病相关联的位点。此问题为全基因组关联性分析(GWAS)问题。题目所给的数据为 500 个健康样本和 500 个患病样本染色体片段上的 9445 个位点编码。

本题探寻的是两个统计变量之间的相关性问题，所以考虑利用统计学中的假设检验方法来进行位点与疾病的关联性分析。在全基因组关联性分析中，并不是检测到的所有 SNP 位点对统计结果有正向的影响，需要在分析之前对 SNP 位点进行质量分析，从而筛选出其中对检验结果有意义的 SNP，剔除其中质量较差会影响统计性能的 SNP。这样不仅可以减少需要参与分析的数据量，同时也可以使统计的结果更有意义。

因此求解本题的思路分为两步，首先对已知的 SNP 位点进行质量筛选，然后利用假设检验的方法分析筛选过后的 SNP 与遗传疾病的相关性。

解决本问题时，需要考虑以下两个问题：

- 如何对已知的所有 SNP 位点进行质量筛选，使筛选的结果更有统计意义；
- 采取何种假设检验的方法能说明检验的结果更加准确。

5.2、SNP 质量筛选

5.2.1、最小等位基因频率（MAF）控制

最小等位基因频率通常是指在给定人群中的不常见的等位基因发生频率，例如某位点有 TT,TC 和 CC 三中基因型，在人群中 C 的频率为 0.28，T 的频率为 0.72，则该位点等位基因 C 的频率为最小等位基因频率，即 $MAF = 0.28$。

在全基因组关联性分析研究中，将 $MAF < 0.01$的 SNP 称为 rare SNP，将 $MAF > 0.05$的称为 commom SNP。*MAF* 值较小的 SNP 会使统计性能降低，从而造成假阴性的结果，因此通常将 $MAF > 0.05$的 SNP 作为首要的研究目标，将 *MAF* 较小的值剔除。

5.2.2、Hardy-Weinberg 平衡控制

根据 Hardy-Weinberg 平衡定律，在理想状态下，各等位基因的频率和等位基因的基因型频率在遗传中是稳定不变的，即保持基因平衡。

在实际检验中，利用卡方统计量来检验某位点的 Hardy-Weinberg 平衡定律是否成立。假设该位点的基因型在所有样本中的频数统计结果如下表：

表 5.1 基因型频数统计结果（观测表）

基因型	AA	Aa	aa	Total
数量	a	b	c	$a+b+c$

此表为观测表，记其中的统计量为 O。由该表，可以计算出每一个等位基因的频率，设 A 的频率为 p，则：$p = \dfrac{2a+b}{2(a+b+c)}$；a 的频率为 $q = 1-p$。

假设该位点符合 Hardy-Weinberg 平衡定律，则期望的基因型-数量表格如下：

表 5.2 基因型频数期望值（期望表）

基因型	AA	Aa	aa	Total
数量	$p^2(a+b+c)$	$2pq(a+b+c)$	$q^2(a+b+c)$	$a+b+c$

此表为期望表，记其中的统计量为 E。

利用卡方统计量来表示这两个表的差异：

$$\chi^2 = \sum \frac{(O-E)^2}{E} \tag{5.1}$$

该统计量的分布符合自由度为 1 的卡方分布。对其差异显著性水平用 *P* 值来表示，*P* 值可以查表得到。

Degrees of freedom (df)	x^2 value[18]										
1	0.004	0.02	0.06	0.15	0.46	1.07	1.64	2.71	3.84	6.64	10.83
2	0.10	0.21	0.45	0.71	1.39	2.41	3.22	4.60	5.99	9.21	13.82
3	0.35	0.58	1.01	1.42	2.37	3.66	4.64	6.25	7.82	11.34	16.27
4	0.71	1.06	1.65	2.20	3.36	4.88	5.99	7.78	9.49	13.28	18.47
5	1.14	1.61	2.34	3.00	4.35	6.06	7.29	9.24	11.07	15.09	20.52
6	1.63	2.20	3.07	3.83	5.35	7.23	8.56	10.64	12.59	16.81	22.46
7	2.17	2.83	3.82	4.67	6.35	8.38	9.80	12.02	14.07	18.48	24.32
8	2.73	3.49	4.59	5.53	7.34	9.52	11.03	13.36	15.51	20.09	26.12
9	3.32	4.17	5.38	6.39	8.34	10.66	12.24	14.68	16.92	21.67	27.88
10	3.94	4.87	6.18	7.27	9.34	11.78	13.44	15.99	18.31	23.21	29.59
P value (Probability)	0.95	0.90	0.80	0.70	0.50	0.30	0.20	0.10	0.05	0.01	0.001

图 5.1 χ^2 分布临界值表

对每一个 SNP 位点进行 Hardy-Weinberg 平衡检验，我们设定一个 p 的阈值，当某位点的 p 值小于该阈值时可以认为该 SNP 位点不符合 Hardy-Weinberg 平衡定律，从而可以将不符合的 SNP 位点剔除。

5.3、模型建立

5.3.1、卡方检验（chi-square test）

卡方检验是一种常用的显著性检验方法，卡方检验可以统计样本的实际观测值与理论（期望）推断值之间的偏离程度。卡方值越大，越不符合原假设；卡方值越小，越符合原假设。统计学中，卡方检验常用来检测两个变量之间的关联程度，此时原假设为两个变量不相关。

本题需要判断位点是否与遗传疾病 A 相关联以及关联程度，因此，针对本问题中的每一个位点，可以考虑分别利用卡方检验来进行数学建模。

提出的原假设为：该位点与遗传疾病 A 无关。我们根据题目所给的 1000 个样本可以进行频率统计得到下面的列联表，即观测表：

表 5.3 1000 样本频率统计列联表

	AA	*Aa*	*aa*	Total
Case(1)	a	b	c	$a+b+c$
Control(0)	d	e	f	$d+e+f$
Total	$a+d$	$b+e$	$c+f$	$a+b+c+d+e+f$

其中的 a,b,c,d,e,f 均由题目所提供的样本数据中统计得到，它们分别表示该列所对应的基因型在该行对应的组中出现的频率，称为观测频数（记为 O）；*Aa*

的含义与问题一中相同，表示编码该该位点的两种不同碱基；Case 表示的为患病样本组，Control 表示的是健康样本组。

在原假设成立的条件下，我们可以根据观测频数计算出上述观测表对应的期望频数（记为E），得到如下期望列联表：

表 5.4 期望列联表

	AA	*Aa*	*aa*	Total
case(1)	$\frac{(a+b+c)(a+d)}{a+b+c+d+e+f}$	$\frac{(a+b+c)(b+e)}{a+b+c+d+e+f}$	$\frac{(a+b+c)(c+f)}{a+b+c+d+e+f}$	$a+b+c$
control(0)	$\frac{(d+e+f)(a+d)}{a+b+c+d+e+f}$	$\frac{(d+e+f)(b+e)}{a+b+c+d+e+f}$	$\frac{(d+e+f)(c+f)}{a+b+c+d+e+f}$	$d+e+f$
Total	$a+d$	$b+e$	$c+f$	$a+b+c+d+e+f$

卡方检验则是比较上述两个列联表的差异，差异越大，则原假设越不成立，即两个变量越相关联。

每个位点的卡方值，也由公式（5.1）指出。该值服从自由度为 2 的χ^2（卡方）分布。其中自由度的计算如下：

$$dif = (R-1)(C-1) = (2-1)(3-1) = 2 \qquad (5.2)$$

其中R表示列联表中参与计算的频数的行数，C代表其列数。

通过查χ^2分布临界值表（图 5.1），可以得到每一个位点的χ^2值所对应的原假设（不相关）成立的几率（P值）。

5.3.2、逻辑回归（Logistic Regression）

针对本问题，每个样本是否患有遗传病可以作为二元的分类问题，因此也可以考虑用逻辑回归模型对每个位点进行数学建模。

令Y_i表示第i个样本的表现型，即是否患病：

$$Y_i = \begin{cases} 0 & \text{健康样本} \\ 1 & \text{病例样本} \end{cases}$$

令X_i表示第i个样本在一个特定位点的基因型编码，此处，我们利用问题一的编码方式，即：

$$X_i = U = \begin{cases} 0 & if\ \ genotype\ is\ AA \\ 1 & if\ \ genotype\ is\ Aa\ or\ aA \\ 2 & if\ \ genotype\ is\ aa \end{cases}$$

对于 9445 个位点中的每一个位点，我们都可以对其建立一个最基本的逻辑回归模型。

令 p_i 表示第 i 个样本患有疾病相对于该位点基因型的条件期望：

$$p_i = E(Y_i = 1 | X_i) \tag{5.3}$$

逻辑回归模型可以表示为：

$$p_i = sigmoid(\beta_0 + \beta_1 X_i) = \frac{1}{1+e^{-(\beta_0+\beta_1 X_i)}} \tag{5.4}$$

$$\text{logit}(p_i) = \ln(\frac{p_i}{1-p_i}) = \beta_0 + \beta_1 X_i \tag{5.5}$$

在此逻辑回归模型中，β_1 值的大小可以粗略的反映两个变量的相关性，β_1 值与 0 的差异越大，说明该位点的基因型与是否患病的相关可能性越大。为了进一步检验两个变量的相关性，在逻辑回归模型中对回归系数 β_1 采用显著性检验的方法求得其 P 值（p-value）。其中最常用的显著性假设方法为 Wald 检验（Wald Test）。

原假设为 $\beta_1 = 0$，表示自变量 X_i 对是否患病的可能性无影响因素。计算统计量：

$$W = \frac{\beta_1^2}{SE_{\beta_1}^2} \tag{5.6}$$

其中 SE_{β_1} 表示 β_1 的标准误差，变量 W 的分布可以近似为自由度为 1 的 χ^2 分布。其 P 值可以查图 5.1 得到。

5.4、问题求解及分析

首先我们分别用最小等位基因频率（MAF）控制和 Hary-Weinberg 平衡控制两项指标对题目所给的 9445 个 SNP 位点进行了 SNP 质量筛选。

最小等位基因频率（MAF）控制筛选时，设置 $MAF > 0.05$，发现所有 SNP 位点均满足，未剔除位点。

Hary-Weinberg 平衡控制筛选时，设置 P 值的阈值为 0.01，其对应的 χ^2 值为 6.64，应当剔除卡方值大于 6.64 的位点。进行筛选之后，我们一共剔除了 97 个 SNP 位点。

接下来，我们分别利用卡方检验和逻辑回归模型两种方法对剩余的 9348 个位点进行了关联性分析。

5.4.1、卡方检验求解

对每个位点，分别在 500 个病例组样例和 500 个对照组样例中统计基因型出现的频率，得到2×3的列联表。利用卡方检验方法，每一个位点都可以得到一个 *P* 值，*P* 值越小，说明该位点基因型与患病相关的可能性越高。

下图以Manhattan图的形式展现了卡方检验模型求解得到的不同位点P值的分布，并对其 *P* 值较小的点进行了坐标标注。

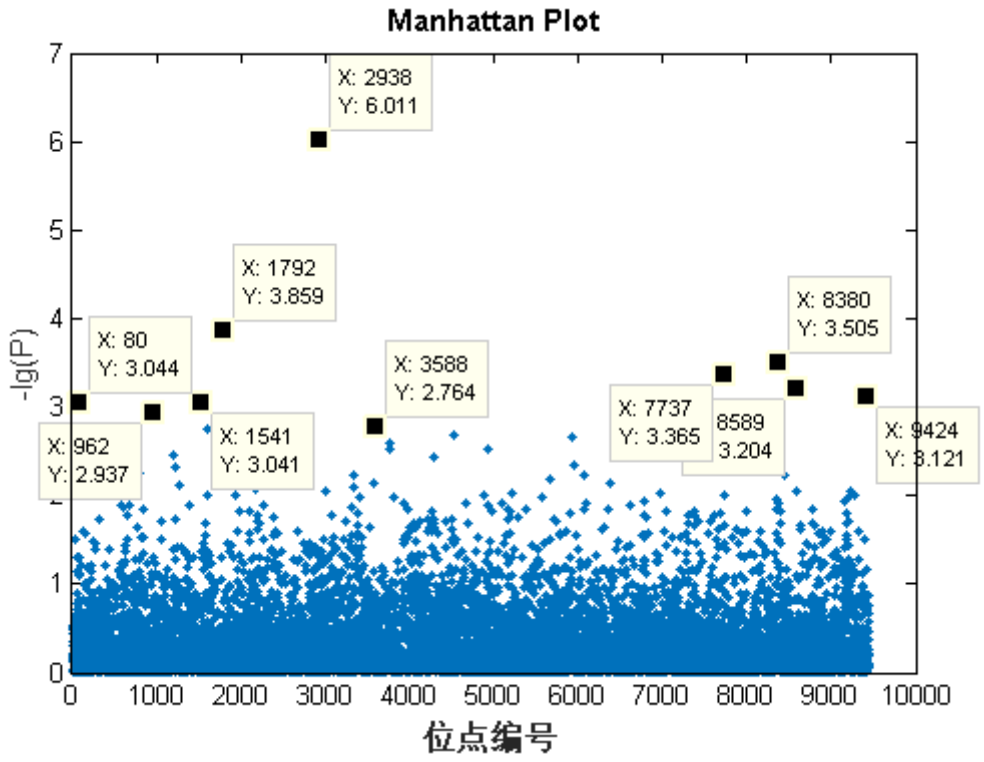

图 5.2 卡方检验模型得到的 Manhattan 图

设置阈值 $P<10^{-4}$，可以找到 8 个相关可能性相对较高的位点，其编号分别为：2938、1792、8380、7737、8589、9424、80、1541。取 *P* 值较小的 5 个位点得到下表。

表 5.5 卡方检验模型得到的 P 值较小的 5 个点

位点编号	2938	1792	8380	7737	8589
位点名称	rs2273298	rs2250358	rs7543405	rs932372	rs9426306
P-value	9.7466×10^{-7}	1.3827×10^{-4}	3.1250×10^{-4}	4.3105×10^{-4}	6.2478×10^{-4}

5.4.2、逻辑回归模型求解

根据问题一中数值编码方式，将题目所给的genotype.dat转化为1000个9445维的数据，每一维对应了一个位点的基因编码，即（ $X_i, i=1,2,3...1000$ ）。

phenotype.txt 分别对应了 1000 个样本的表现型，即（$Y_i, i=1,2,3...1000$）。

我们利用 MATLAB 工具 glmfit()对逻辑回归模型进行求解，得到每一个位点的 β_1 和 P 值。其对应的 P 值越小说明该位点与患病的相关可能性越高。

下图以 Manhattan 图的形式展现了逻辑回归模型求解得到的不同位点 P 值的分布，并对其 P 值较小的点进行了坐标标注。

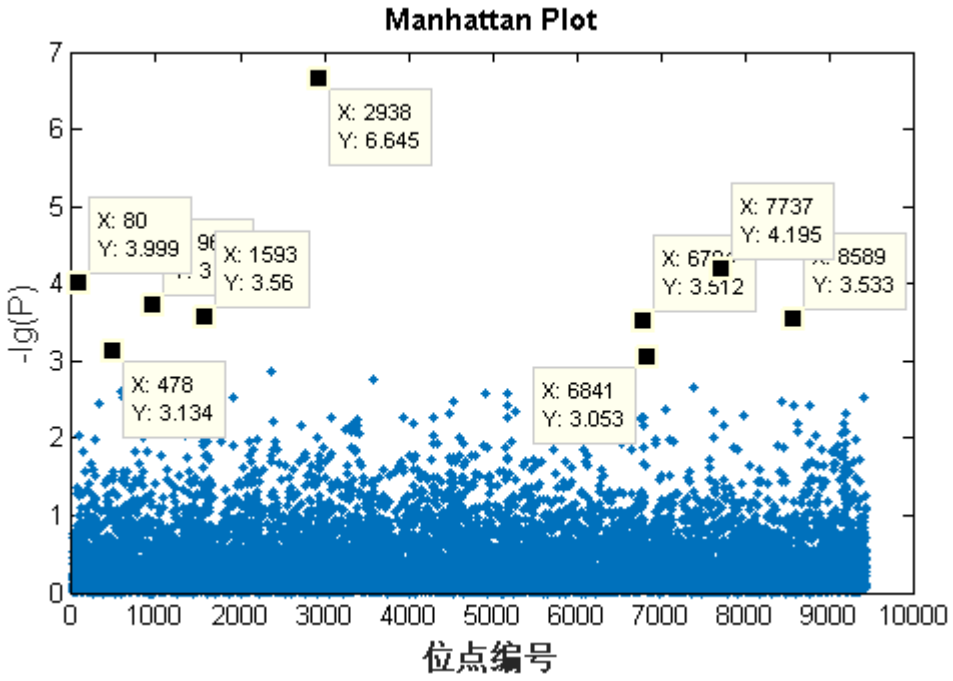

图 5.3 逻辑回归模型得到的 Manhattan 图

设置阈值 $P<10^{-4}$，可以找到 9 个相关可能性相对较高的位点，其编号分别为：2938、7737、80、962、1593、8595、6794、478、6841。取 P 值较小的 5 个位点得到下表。

表 5.6 逻辑回归模型得到的 P 值较小的 5 个点

位点编号	2938	7737	80	962	1593
位点名称	rs2273298	rs932372	rs12036216	rs4391636	rs7522344
P-value	2.2654×10^{-7}	6.3761×10^{-5}	1.0024×10^{-4}	1.8641×10^{-4}	2.7555×10^{-4}

5.5、求解分析及评价

在解决本题时，我们首先对题目所给数据进行了预处理，采用 MAF 和 Hary-Weinberg 控制两种方法，对待检验的 9445 个 SNP 位点进行了质量筛选，剔除其中质量不符合要求的 SNP 位点，这样的预处理方法不仅可以提高假设检验时的统计性能，还可以降低运算的复杂性。

接下来，我们分别用两种模型来对 SNP 和患病性状的相关性进行建模，可以发现，两种模型找到的与该疾病比较相关的共同的位点是第 2938 号位点（rs2273298）和第 7737 号位点（rs932372），最为相关的位点是第 2938 号位点（rs2273298），另外，也可以发现，两种模型中，第 80 号位点（rs12036216）和第 8589 号位点（rs9426306）的 P-value 也都比较小，综合以上两种模型的结果，我们可以认为，与该疾病最有可能相关的位点是第 2938 号位点（rs2273298），其次是第 7737 号位点（rs932372），而第 80 号位点（rs12036216）和第 8589 号位点（rs9426306）也存在一定的相关性。

通过以上的分析可以发现，如果我们仅仅依靠卡方检验或逻辑回归中的一种模型，那么得到的结果就很有可能有比较大的偏差，而找寻这两种模型统计结果的一致性，就可以避免一种假设检验的特殊性，从而提高结果的可信度。所以本题中，采用两种模型共同验证、互为对照的方式是比较科学有效的。

我们也设想过性状之间可能会有相关性，这种相关性也有可能对性状产生一定的影响，但是由于计算这种相关关系会产生比较大的运算量，受限于时间和计算机性能，我们并没有做出特别深入的研究。

6、问题三模型建立与求解

6.1、问题描述及分析

本题的目的是设计一种方法，找出与某遗传疾病相关联的基因。题目在问题二所给的数据外还提供了 300 个基因所包含的位点组成，将基因作为位点的集合。

本题探寻的问题与问题二类似，也是找出两个统计变量之间的相关性。所以也可以考虑利用统计学中的假设检验方法来分析。与问题二不同的是问题二探究的对象为单个 SNP 位点，而本题的探究的对象是基因。由题意，基因可以看做若干位点的集合。由该集合中的全集或子集位点共同影响该基因与疾病之间的关系。

本题要求我们寻找与遗传疾病 A 关联性最强的一个或几个致病基因，比较自然的想法自然是根据第二问里面找出的致病位点，从文件夹 geno_info 中的基因与位点之间的映射关系得到最有可能致病的基因。但是这样会有两个问题，一是最有可能致病的位点所在的基因未必是最有可能致病的基因；二是得到致病位点的过程中已经使用了假设检验，根据致病位点与基因之间的从属关系选取基因可能在二次假设检验的时候会引入更多的不确定性。因此我们不使用第二题中得到的结果，而是将每个基因作为一个其包含的位点的集合来看待。从而可以考虑两种统计方法：逻辑回归（Logistic Regression）和 SKAT(SNP-set Kernel Association Test)。其中 Logistic Regression 方法未能揭示位点之间的交互关系，而 SKAT 模型本身就考虑进了位点之间的交互关系，因此结果会更有说服力。

6.2、模型建立

6.2.1、逻辑回归（Logistic Regression）模型

与问题二类似，该问题仍为一个二元分类问题。因变量仍为样本的表现型，即是否患有疾病：

$$Y_i=\begin{cases}0 & \text{健康样本}\\ 1 & \text{病例样本}\end{cases}$$

对于一个特定的基因，假设它为 M 个 SNP 位点的集合。则对于一个特定的基因，我们令 $X_i^{(m)}$ 表示第 i 个样本在该基因中第 m 个位点的基因型编码，此处，仍旧利用问题一的编码方式，即：

$$X_i^{(m)}=U=\begin{cases}0 & if\ \ genotype\ is\ AA\\ 1 & if\ \ genotype\ is\ Aa\ or\ aA\\ 2 & if\ \ genotype\ is\ aa\end{cases}$$

对于题目所给的 300 个基因中的每一个基因，我们都可以对其建立一个回归模型。

令 p_i 表示第 i 个样本患有疾病相对于该位点基因型的条件期望：

$$p_i=E(Y_i=1\mid X_i^{(1)},X_i^{(2)},...,X_i^{(M)}) \qquad (6.1)$$

逻辑回归模型可以表示为：

$$p_i=sigmoid(\beta_0+\beta_1X_i^{(1)}+...+\beta_MX_i^{(M)})=\frac{1}{1+e^{-(\beta_0+\beta_1X_i^{(1)}+...+\beta_MX_i^{(M)})}} \qquad (6.2)$$

$$\text{logit}(p_i)=\ln(\frac{p_i}{1-p_i})=\beta_0+\beta_1X_i^{(1)}+...+\beta_MX_i^{(M)} \qquad (6.3)$$

其中，假设 M 个自变量为独立变量。这个模型相对于我们在问题二中建立的逻辑回归模型更具有一般性。并且考虑了多个位点的共同影响。

在此逻辑回归模型中，β_m 值的大小可以粗略的反映该基因中第 m 个位点与患病的相关性。而我们需要考虑的是整个基因的联合共同作用，而不是单位点的影响，因此为了进一步检验基因与患病的相关性，在逻辑回归模型中对所有自变量的回归系数综合采用显著性检验的方法求得其 P 值（p-value）。显著性检验仍采用 Wald Test。

此时，原假设为 $\beta_1=\beta_2=\cdots=\beta_M=0$，表示该基因（其中的所有位点）对是否患病的可能性无影响因素。计算统计量：

$$W=\sum_{m=1}^{M}\frac{\beta_m{}^2}{SE_{\beta_m}^2} \qquad (6.4)$$

其中 SE_{β_m} 表示 β_m 的标准误差，变量 W 的分布可以近似为自由度为 M 的 χ^2 分布。由此，我们可以得到每一个基因的综合的 P 值。P 值越小，该基因与患病相关的可能性越大。

6.2.3、SKAT（Sequence Kernel Association Tests）检验

在全基因组关联性分析中，SKAT 检验是一个基因级别的检验方法。

首先，我们考虑对每一个基因建立一个回归模型，令 $\mathbf{Y}$ 表示样本的表现型（是否患病），$\mathbf{X}$ 表示一个 $N \times M$ 的样本基因型矩阵，其中 N 为样本的个数，M 为组成该基因的位点数，对于每一个样本，回归模型表示如下：

$$g[E(Y_i)] = \alpha_0 + \mathbf{C_i}\boldsymbol{\alpha} + \mathbf{X_i}\boldsymbol{\beta} \tag{6.5}$$

其中，$\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_p)^T$ 表示协变量的回归系数，$\mathbf{C}_i = (C_{i1}, ..., C_{ip})$ 表示回归模型中的协变量，$\mathbf{X}_i = (X_{i1}, ..., X_{iM})$ 表示第 i 个样本的位点基因型向量，Y_i 表示第 i 个样本是否患病，$\boldsymbol{\beta} = (\beta_1, ..., \beta_M)^T$ 表示 M 个位点基因型的回归系数。

我们对 β 进行一些假设：$\begin{cases} E(\beta_j) = 0 \\ Var(\beta_j) = w_j^2 \tau \\ corr(\beta_j, \beta_k) = \rho \quad j \neq k \end{cases}$

在进行关联性检验的时候，原假设依然为：$\beta = 0$，表示该基因对患病与否无影响。定义一个得分统计量（score statistic）：

$$Q_\rho = (\mathbf{Y} - \hat{\boldsymbol{\mu}}_\mathbf{0})^T \mathbf{K}_\rho (\mathbf{Y} - \hat{\boldsymbol{\mu}}_\mathbf{0}) \tag{6.6}$$

$$\mathbf{K}_\rho = \mathbf{XWR}_\rho \mathbf{WX}^\mathbf{T} \tag{6.7}$$

其中 $\mathbf{R}_\rho = (1-\rho)\mathbf{I} + \rho \mathbf{1}\mathbf{1}^T$ 表示相关矩阵，$\mathbf{W} = \mathrm{diag}(w_1, ..., w_M)$。$\hat{\boldsymbol{\mu}}_\mathbf{0}$ 表示了在模型原假设成立的情况下，观测量 $\mathbf{Y}$ 的期望值。

SKAT 检验模型在上述一般性模型的基础上进行了 $\rho = 0$ 的约束，由此我们的到 SKAT 模型的得分统计量为：

$$Q_{\rho=0} = \sum_{j=1}^{M} w_j^2 [\sum_{i=1}^{N} (Y_i - \hat{\mu}_{i,0}) X_{ij}]^2 \tag{6.8}$$

其中 $w_j = Beta(\hat{p}_j, 1, 25)$，$\hat{p}_j$ 为第 j 个位点的最小等位基因频率（MAF）估计值。

上述统计量 $Q_{\rho=0}$ 近似服从自由度为 1 的卡方分布（χ_1^2）。

6.3、模型求解

6.3.1、逻辑回归模型求解

根据问题一中数值编码方式，将题目所给的 300 个基因用 m 个位点编码 phenotype.txt 分别对应了 1000 个样本的表现型，即（$Y_i, i = 1, 2, 3...1000$）。

我们利用 MATLAB 工具 fitglm()对逻辑回归模型进行求解，得到每个基因各位点的联合 *P* 值。其对应的 *P* 值越小说明该位点与患病的相关可能性越高。

下图以 Manhattan 图的形式展现了逻辑回归模型求解得到的不同基因 *P* 值的分布，并对其 *P* 值较小的点进行了坐标标注。

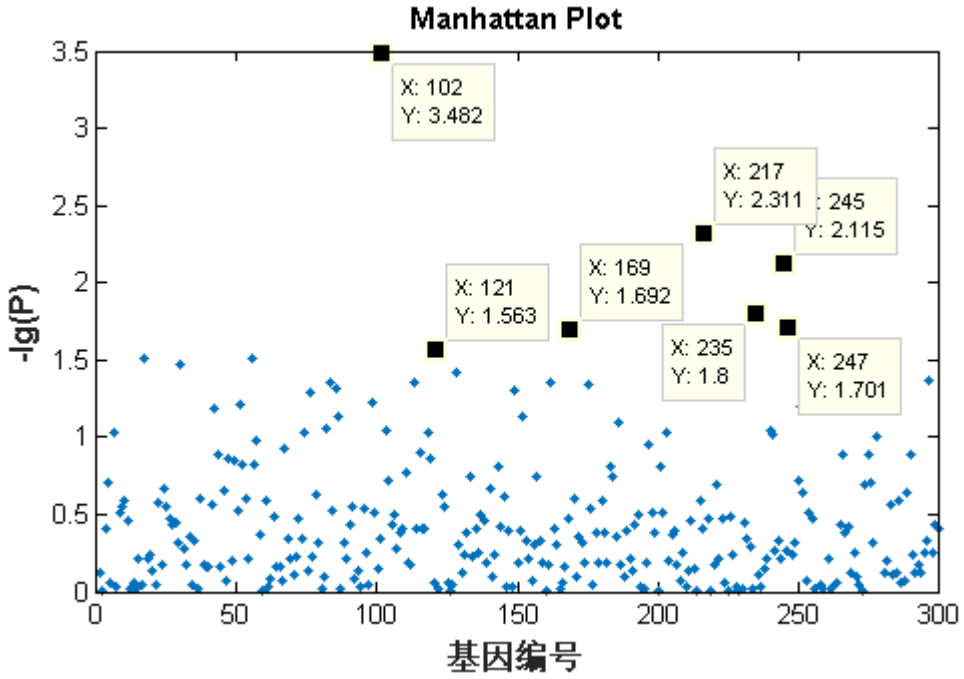

图 6.1 逻辑回归模型得到的 Manhattan 图

观察此图可以发现由逻辑回归模型检验出的与遗传疾病 A 最相关的基因为 102 号基因，其次为第 217 号基因。

6.3.2、SKAT 检验模型求解

SKAT算法有R语言的开源的实现：PACKAGE‘SKAT’，我们进行算法求解使用的编程语言是MATLAB和R语言，其中MATLAB主要用于数据处理部分。我们首先要对数据进行预处理，来满足SKAT所需要的格式。对于我们的应用，需要的数据只有表示是否患病的向量 **Y** 和根据第一题编码方式得到的数据矩阵 **X**（该矩阵元素只包含0,1,2三个数值），这些数据可以直接使用MATLAB内置的函数dlmwrite来写入文件。在使用R语言自带的函数读取文件后，调用以下的两个函数即可求解SKAT模型:

```
obj<-SKAT_Null_Model(y ~ 1, out_type="D")
SKAT(Z, obj, kernel = "linear.weighted")$p.value.
```

同样的，下图以 Manhattan 图的形式展现了 SKAT 检验模型求解得到的不同基因 *P* 值的分布，并对其 *P* 值较小的点进行了坐标标注。

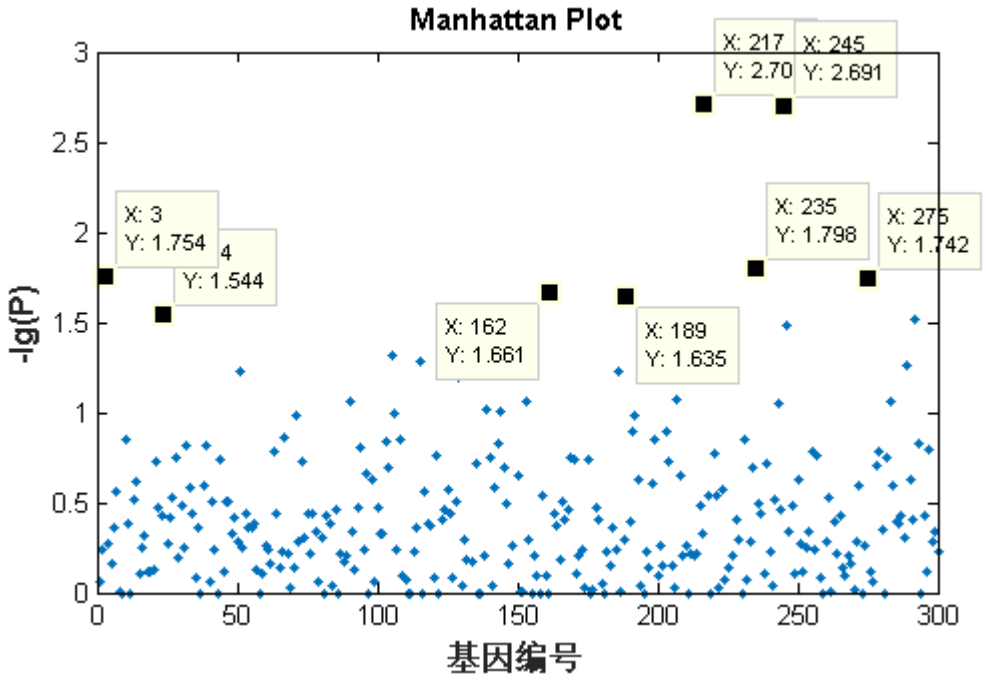

图 6.2 SKAT 检验模型得到的 Manhattan 图

观察此图可以发现由 SKAT 检验模型检验出的与遗传疾病 A 最相关的基因为 217 号基因，其次为第 245 号基因。

6.4、求解分析及评价：

我们解决本题主要使用了两种模型：逻辑回归模型（Logistic Regression）和 SKAT 模型。Logistic Regression 中得到的与该疾病关联性最大的基因是第 102 号基因，该基因包含十个位点；SKAT 模型解得的关联性最大的基因是第 217 号基因，该基因包含了二十个位点。

结合问题二的检验结果分析，组成第 102 号和 217 号基因的位点及其在问题二中算出的 p-value 分别如下面两个表格所示：

表 6.1 第 102 号基因上面的十个位点及其各自的 p-value

位点名	rs12144133	rs6696978	rs2273298	rs2273299	rs6667049
p-value	0.1863	0.0571	9.7466e-07	0.2147	0.2605
位点名	rs12092513	rs9783053	rs12074936	rs2180184	rs6541080
p-value	0.3895	0.4264	0.7143	0.5136	0.0942

表 6.2 第 217 号基因上的二十个位点及其 p-value

位点名	rs2870446	rs2744720	rs17356059	rs2807349	rs2807347
p-value	0.4649	0.1205	0.0289	0.2688	0.1424
位点名	rs3765340	rs2473246	rs7513455	rs2473247	rs17356087
p-value	0.0605	0.0113	0.0515	0.0561	0.4216
位点名	rs2473252	rs2473253	rs12080095	rs2744728	rs2505722
p-value	0.1117	0.0280	0.5678	0.9357	0.0169
位点名	rs2807345	rs10917176	rs7552560	rs1569583	rs2744731
p-value	0.0022	0.0744	0.1557	0.1921	0.8645

可以发现第二题中预测的最有可能与患病有关的位点 rs2273298（编号为 2938 号）恰好位于第 102 号基因上。这一发现似乎印证了 Logistic Regression 结果的正确性，但是由表 6.1 可知，第 102 号基因中除了位点 rs2273298 之外的其它位点显著性水平并不理想，仅凭观察不能确定它比 217 号基因的 *P* 值小，rs2273298 位点位于该基因上可能只是巧合，正是由此，才导致 Logistic Regression 模型假设中，该基因的 *P* 值显著，而不是该基因位点的集合导致 *P* 显著。此外，Logistic Regression 模型假定了位点之间的相互独立性，这在很多情况下是不成立甚至严重偏离现实的（位点之间通常存在相互关联），通过对比由两种方法得到结果的 Manhattan 图，可见 SAKT 检验模型的 Manhattan 图中，显著性较高的基因在 Logistic Regression 模型的 Manhattan 图中的显著性也较高，反之则不然，比如 102 号基因。综上，第 217 号基因或者第 245 号基因是与疾病 A 关联性最强的基因更有说服力。

7、问题四模型建立与求解

7.1、问题描述及分析

本题的目的在于检验遗传位点与多个性状之间的关系。题目在问题二提供的 1000 个样本的 9445 个位点信息之外还提供了这 1000 个样本的 10 个相关联性状的信息。

本题本质上是考察多个遗传位点同多个性状之间的关系。我们首先考虑这 10 个性状之间的相互关联，希望能够对 10 个性状进行分组，再分别对每组性状研究与其相关联的 SNP 位点。首先尝试了对性状两两之间进行卡方检验，得到了如下的结果：

表 7.1 两两性状之间的卡方统计量

i / j	1	2	3	4	5	6	7	8	9	10
1	0	512.65	547.60	501.26	467.86	462.40	547.60	506.94	553.54	529.98
2		0	506.94	473.34	462.40	478.86	512.66	535.82	541.69	559.50
3			0	490.00	524.18	478.86	524.18	529.98	506.94	559.50
4				0	467.86	456.98	518.40	535.82	518.40	506.94
5					0	559.50	462.40	501.26	446.22	478.86
6						0	467.86	501.26	456.98	446.224
7							0	541.70	518.40	577.60
8								0	529.98	577.60
9									0	553.54
10										0

上表中(i,j)数值代表第 i 个性状与第 j 个性状之间的卡方统计量，卡方统计值均非常大，由此可见这些性状两两之间都具有极强的相关性；我们还使用了余弦距离来度量性状之间的相似性，发现性状两两之间的相似性数值接近，因此难以根据性状之间是否相似来对性状进行分组；同样，我们也不能使用普通的聚类算法（如 kmeans），因为样本数量只有 10 个，但是样本维度高达 1000 维，在这样高的维度下，样本空间会变得非常稀疏，两个样本之间的欧氏距离无法精确的衡量它们的相似性，所以简单的聚类会造成严重的维度灾难，需要思考其他的做法。

我们还考虑使用典型相关分析(Canonical Correlation Analysis)，但是该方法一般需要个体之间的相似性信息。因此，我们需要考虑一种方法可以避免以上出现的维灾以及无法得到个体间相似性信息的问题。

7.2、模型建立

7.2.1、metaCCA 算法

我们主要使用了文献[1]中提出的 metaCCA 算法，该算法克服了传统 CCA 的需要个体之间的相似性信息这一问题并使用协方差矩阵收缩来增强其鲁棒性。

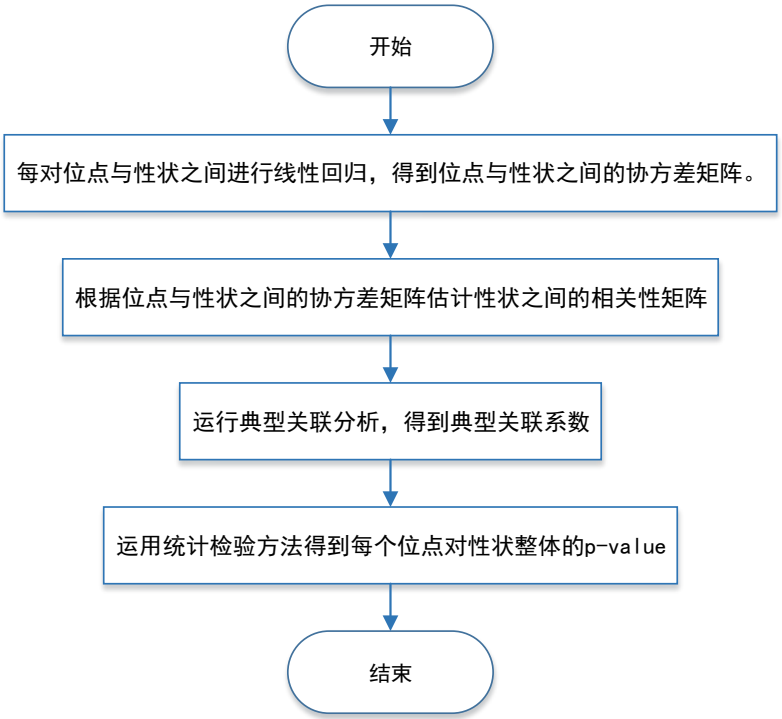

图 7.1 metaCCA 算法的流程图

上述流程图简要描述了 metaCCA 算法的步骤。首先进行利用线性回归得出位点与性状之间的协方差矩阵；再以此估计性状之间的相关矩阵；然后进行典型关联分析的到关联系数；最后利用统计检验的方法求得每个位点对性状整体的 P 值。下面介绍其数学理论建模基础。

用 $\mathbf{X}$ 和 $\mathbf{Y}$ 分别代表基因型和性状的矩阵，假设它们的大小分别为 $N\times P$ 和 $G\times P$，N 是样本数量，在本题中为 1000，G 和 P 分别为位点数量和性状数量，

在本题中为 9445 和 10。

一般的全基因组关联性分析(GWAS)中，对于单个位点 $x_g \in R^N$ 和单个性状 $y_p \in R^N$ 之间的建模可以简单地使用线性回归：

$$y_p = \alpha_{gp} + x_g \beta_{gp} + \varepsilon \tag{7.1}$$

系数 β_{gp} 为回归直线的斜率，描述了基因位点 x_g 对于性状 y_p 的影响程度。α_{gp} 代表偏移量，ε 代表高斯噪声项，通过最小平方和误差可以得到对 β_{gp} 的闭式估计：

$$\beta_{gp} = [x_g^T y_p][x_g^T x_g]^{-1} = [(N-1)s_{xy}][(N-1)s_{xx}]^{-1} = s_{xy} \tag{7.2}$$

其中的 s_{xy} 代表位点 x_g 和表现型 y_p 之间的样本协方差，$s_{xx} = 1$ 代表 x_g 的样本方差(假设已经被归一化了)，于是全部位点与性状之间的协方差矩阵可以由线性回归系数 β_{gp} 来表示：

$$\Sigma_{XY} = \begin{pmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{1P} \\ \beta_{21} & \beta_{22} & \cdots & \beta_{2P} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{G1} & \beta_{G2} & \cdots & \beta_{GP} \end{pmatrix} \tag{7.3}$$

性状之间的协方差矩阵 Σ_{YY} 可以根据 Σ_{XY} 进行估计，因为 Σ_{YY} 的每一个元素对应于 Σ_{XY} 两列之间的皮尔逊相关系数。对于表现型 s 和 t 而言：

$$\Sigma_{YY}(s,t) = \frac{\sum_{g=1}^{G}(\beta_{gs} - \mu_s)(\beta_{gt} - \mu_t)}{\sqrt{\sum_{g=1}^{G}(\beta_{gs} - \mu_s)^2}\sqrt{\sum_{g=1}^{G}(\beta_{gt} - \mu_t)^2}} \tag{7.4}$$

其中 $\mu_s = \frac{1}{G}\sum_{g=1}^{G}\beta_{gs}$，$\mu_t = \frac{1}{G}\sum_{g=1}^{G}\beta_{gt}$ 为性状 s 和 t 对应列的均值，而位点之间的协方差矩阵 Σ_{XX} 一般从位点数据库得到，本题中未给出故使用单位矩阵 $\Sigma_{XX} = I$。

CCA 是一种检测两组变量 $X \in R^{N\times G}$ 和 $Y \in R^{N\times P}$ 之间线性关系的技术，其中 X 和 Y 一般是关于同样对象的两组特征。目标函数是寻找两个矩阵列之间线性组合的极大值，这等价于寻找向量 $a \in R^G$ 和 $b \in R^P$ 使得目标函数：

$$r=\frac{(Xa)^T(Yb)}{\|Xa\|\bullet\|Yb\|}=\frac{a^T\Sigma_{XY}b}{\sqrt{a^T\Sigma_{XX}a}\sqrt{b^T\Sigma_{YY}b}} \tag{7.5}$$

极大化。相关性系数r的极大值被称为X和Y之间的典型关联。

为了求取a和b，需要先求得矩阵K：

$$K=\Sigma_{XX}^{-1/2}\Sigma_{XY}\Sigma_{YY}^{-1/2} \tag{7.6}$$

对K进行奇异值分解，可得：

$$K=(\alpha_1,\ldots,\alpha_k)D(\gamma_1,\ldots,\gamma_k)^T \tag{7.7}$$

式(7.7)中，α_l和γ_l分别代表KK^T和K^TK经过正规化的特征向量，D是对角矩阵且对角线元素为对应的特征值的算术平方根，$D=diag(\sqrt{\lambda_1},\ldots,\sqrt{\lambda_k})$于是典型关联系数可依下式算出：

$$r_l=\sqrt{\lambda_l} \tag{7.8}$$

关联权重可以根据α_l和γ_l算得：

$$\alpha_l=\Sigma_{XX}^{-1/2}\alpha_l,\beta_l=\Sigma_{YY}^{-1/2}\gamma_l \tag{7.9}$$

可以根据卡方检验来调查典型关联分析的统计显著性，测试分为两步进行。第一步，根据特征值计算以下的统计量：

$$\Lambda_l=\prod_{i=l}^{k}(1-\lambda_i) \tag{7.10}$$

随后，下式的右端近似服从卡方分布：

$$\chi_l^2=-[(N-1)-\frac{G+P+1}{2}]\ln\Lambda_l \tag{7.11}$$

式中N是X和Y的样本数量，该卡方分布的自由度为$(G-l+1)\times(P-l+1)$，我们在问题求解中使用的卡方统计量为χ_1^2。

7.3、模型求解：

metaCCA 算法的作者发布了该算法的 MATLAB 实现，我们进行算法求解使用的编程语言是 MATLAB。我们首先要对数据进行预处理，来满足 metaCCA 所需要的格式，metaCCA 需要的输入只有一个Σ_{XY}，即位点与表现型之间的协方差矩阵，这个矩阵要求被事先归一化过，因此我们首先调用 MATLAB 的 fitlm 函数，得到协方差矩阵Σ_{XY}；在对其归一化以后根据协方差矩阵的性质估算Σ_{YY}；最后调用函数：

```
metaCCA_result = metaCCA(2, S_XY, S_XY,...
            0, 0, S_YY, S_YY, N1, N2 );
```

可以计算得出典型分量系数和 p-value。

下图以 Manhattan 图的形式展现了 metaCCA 模型求解得到的不同位点 *P* 值的分布，并对其 *P* 值较小的点进行了坐标标注。可以发现，该模型求解得到的第 4569 号位点的显著性明显高于其他位点。

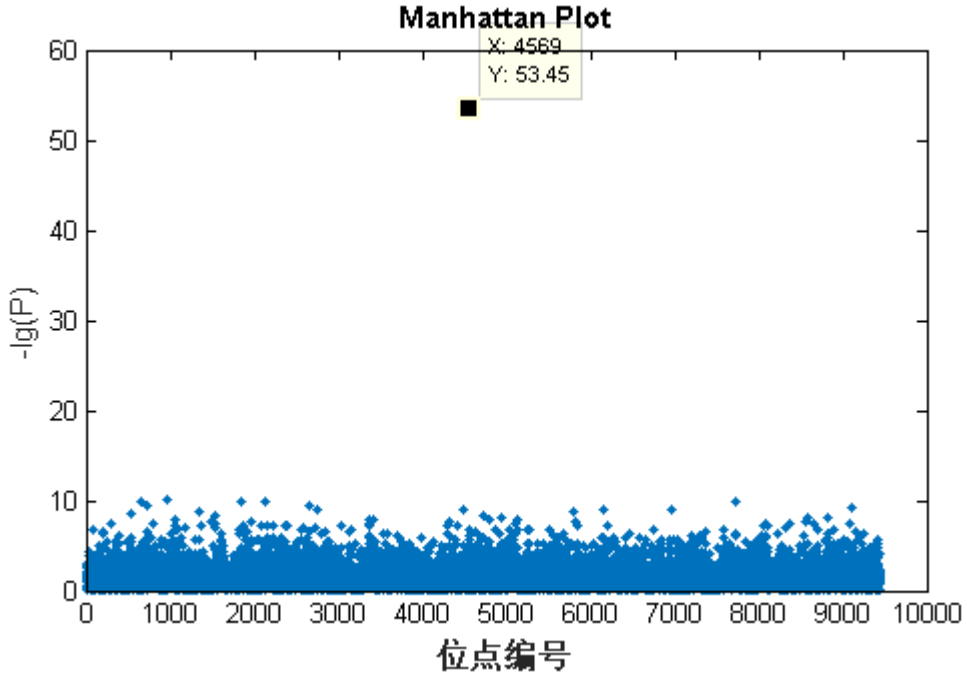

图 7.2 metaCCA 模型得到的 Manhattan 图

7.4、求解分析及评价：

解决本题我们使用了 metaCCA 算法，最后找到的位点是 rs12746773，即第 4569 个位点，其对应的 p-value $\approx 10^{-53}$，关联系数为 $r = 0.3617$，次优的位点是 rs10915423，即第 966 个位点，对应的 p-value $\approx 10^{-10}$，关联系数为 $r = 0.1834$。P-value 和关联系数之间差距巨大，说明该位点与这十个性状之间具有很强的线性关联性。即与这 10 个性状关联性最强的位点是 rs12746773。

8、模型总结与评价

本文研究的是人类基因或位点和性状或疾病的相关性问题，根据现有的 1000 个样本的疾病信息、样本的 9445 个位点编码信息，以及包含这些位点的基因信息，采用卡方检验和逻辑回归模型、SKAT 模型及 metaCCA 模型等多种方式建立数学模型，通过 Matlab、R 语言等工具求解，最终确定了与疾病或相

关性状相关性较强的致病基因或致病位点。

本模型的优点有：

1. 在进行性状和 SNP 位点相关性分析之前，采用最小等位基因频率（MAF）控制和 Hary-Weinberg 平衡控制两种方式，先对 SNP 位点进行质量筛选，剔除质量不合格的数据，使筛选后的结果更有统计意义，同时也减小了运算量；
2. 在发现的致病位点或基因过程中，同时建立多种模型独立进行计算，各模型的结果互为对照，互相映证，综合考虑，提高了分析结果的可靠性和准备性；
3. 模型的扩展性和可移植性比较强，当题目中样本的性状和位点的数量和复杂性发生变化时，此模型仍然适用。

本文在分析性状和位点或基因的相关性时做了深入的探索和改进的工作，但是现有的工作依然有待完善之处，今后应继续改进和深入探索的内容如下：本文没有深入分析位点与位点之间的相互影响关系，以及多个相互影响的位点对性状的共同作用，由于计算位点之间的相关性的运算量过于庞大，受限于计算机的计算性能，需要考虑更加高效的方法来实现，因为竞赛时间有限，此项工作将在日后的工作学习中继续进行。

参考文献

[1] Ionita-Laza I, Lee S, Makarov V, et al. Sequence kernel association tests for the combined effect of rare and common variants[J]. The American Journal of Human Genetics, 2013, 92(6): 841-853.

[2] Montana G. Statistical methods in genetics[J]. Briefings in bioinformatics, 2006, 7(3): 297-308.

[3] Lehne B, Lewis C M, Schlitt T. From SNPs to genes: disease association at the gene level[J]. PloS one, 2011, 6(6): e20133.

[4] Jim Stankovich. Statistical analysis of genome-wide association (GWAS) data[M]. Menzies Research Institute University of Tasmania J, 2014.

[5] Crosses E. Review of statistical methods for QTL mapping in experimental crosses[J]. Lab animal, 2001, 30(7).

[6] Li, B., and Leal, S.M. (2008). Methods for detecting associations with rare variants for common diseases: application to analysis of sequence data. Am. J. Hum. Genet. 83, 311–321.

[7] Lin, D.Y., and Tang, Z.Z. (2011). A general framework for detecting disease associations with rare variants in sequencing studies. Am. J. Hum. Genet. 89, 354–367.

[8] Wang WYS, Barratt BJ, Clayton DG, et al. Genome-wide association studies: theoretical and practical concerns. Nat Rev Genet 2005;6:109–18.

[9] Cui Y, Kang G, Sun K, et al. Gene-centric genomewide association study via entropy[J]. Genetics, 2008, 179(1): 637-650.

附录

Problem 2-卡方检验:

```
data = importdata('genotype.dat');
genodata = zeros(1000, 9445);
genotype = regexp(data{1},'\s+','split');
temp = cell(1000, 9445);
for i = 2 : 1001
    temp(i-1,:) = regexp(data{i},'\s+','split');
end
for i = 1 : 9445
    t = sort(unique(temp(:,i)));
    if (numel(find(strcmp(temp(:,i), t(1))))< ...
            numel(find(strcmp(temp(:,i), t(3)))))
        tmp = t{1};
        t{1} = t{3};
        t{3} = tmp;
    end
    genodata(find(strcmp(temp(:,i),t(1))),i) = 0;
    genodata(find(strcmp(temp(:,i),t(2))),i) = 1;
    genodata(find(strcmp(temp(:,i),t(3))),i) = 2;
end
save geno genotype genodata
chi = zeros(2,3);
r = zeros(1, size(genodata, 2));
for i = 1 : size(genodata, 2)
    for j = 0 : 2
        chi(1,j+1) = numel(find(genodata(1:500,i)==j));
        chi(2,j+1) = numel(find(genodata(501:1000,i)==j));
    end
    r(i) =
sum(sum((abs(chi(:,:)-repmat(sum(chi(:,:))/2,[2,1]))-.5).^2./...
        repmat(sum(chi(:,:))/2,[2,1])));
end
pval = 1 - chi2cdf(r, 2);
save problem2 pval
```

Problem 2-逻辑回归：

```
% This is the program for problem2
% usage: this file contains many parts, you can run every part at once
% by removing or
% commenting other parts, pay attention not to run one part for many times!


% part1:load origin data file(genotypetype.dat) and convert it to cell
format
mat=cell(1000,9445);
for i=1:1000,
    str=char(genotype_data(i));
    mat(i,:)=regexp(str,' ','split');
end
% part1 end


% initial
p_value=zeros(9445,1);
rs_str={'AA','CC','GG','TT','AC','AG','AT','CA','CG','CT',...
'GA','GC','GT','TA','TC','TG'};
% part2:encode the origin data to 0 1 2
for j=1:9445,
    temp1=17;
    temp2=17;

    for i=1:1000,
        for k=1:16,
            if strcmp(mat(i,j),rs_str(k)) == 1
                if k<5
                    if temp1==17
                        temp1=k;
                        break;
                    elseif k~=temp1
                        temp2=k;
                        break;
                    end
                end
            end
        end
        if temp2 ~= 17
            break;
        end
    end
```

```
    if temp1>temp2
        temp1=temp2;
    end

    for i=1:1000,
        for k=1:16,
            if strcmp(mat(i,j),rs_str(k)) == 1
                if k==temp1
                    mat(i,j)=num2cell(0);
                elseif k>4
                    mat(i,j)=num2cell(1);
                else
                    mat(i,j)=num2cell(2);
                end
            end
        end
    end
end
% part2 end


% part3:remove the dirty data using WAF
for i=1:9445,
    [a,~] = size(find(cell2mat(mat(:,i))==0));
    [b,~] = size(find(cell2mat(mat(:,i))==1));
    [c,~] = size(find(cell2mat(mat(:,i))==2));
    % if p_value is less than 0.05, it is dirty
    if a<c
        p_value(i,1)=(2*a+b)/2/(a+b+c);
    else
        p_value(i,1)=(2*c+b)/2/(a+b+c);
    end
end
% part3 end


% part4:remove the dirty data using Hary-Weinberg
chi=zeros(9445,1);
for i=1:9445,
    [a,~] = size(find(cell2mat(mat(:,i))==0));
    [b,~] = size(find(cell2mat(mat(:,i))==1));
    [c,~] = size(find(cell2mat(mat(:,i))==2));
    ex_a = ((2*a+b)^2)/(4*(a+b+c));
```

```
    ex_b = (a+b+c)-((2*a+b)^2)/4/(a+b+c)-((2*c+b)^2)/4/(a+b+c);
    ex_c = ((2*c+b)^2)/4/(a+b+c);
    % if chi value is more than 6.64, it is dirty
    chi(i,1) = (a-ex_a)^2 / ex_a + (b-ex_b)^2 / ex_b + (c-ex_c)^2 / ex_c;
end
% part4 end


% part5: logit
beta1=zeros(9445,1);
beta0=zeros(9445,1);
for j=1:9445,
    [theta,~,s]=glmfit(cell2mat(mat(:,j)),[phenotype
ones(1000,1)],'binomial','link','logit');
    beta0(j,1)=theta(1,1);
    beta1(j,1)=theta(2,1);
    p_value (j,1)=s.p(2);
end
% part5 end


% part6: plot using manhatten
manhattan=zeros(9445,1);
for i=1:9445,
    manhattan(i,1)=-log10(p_value(i,1));
end
plot(manhattan,'.');
%part6 end
```

Problem3-逻辑回归：

```
% This is program for problem3

% load origin data file(gene*.dat) and convert it to cell format
for i=1:300,
    gene_name=importdata(['gene_info\gene_',num2str(i),'.dat']);
    [m,n]=size(gene_name);
    for a=1:m,
     for b=1:9445,
         if strcmp(S(1,b),gene_name(a,1)) == 1
             g{1,i}(:,a)=num2cell(mat(:,b));
             break;
         end
     end
    end
end

for j=1:300,
    [m,n]=size(g{1,j});
    for a=1:n,
        for i=1:1000,
            gene{1,j}(i,a)=cell2mat(g{1,j}{i,a});
        end
    end
    gene{1,j}(:,n+1)=phenotype(:,1);
end

% convert to dataset
gene=cell(1,300);
for i=1:300,
    [m,n]=size(g{1,i});
    gene{1,i}=zeros(1000,n+1);
end

gene_ds=cell(1,300);
for i=1:300,
    gene_ds{1,i}=mat2dataset(gene{1,i});
end


% caculate the all p-value
gene_mdl=cell(1,300);
for i=1:300,
    [m,n]=size(gene{1,i});
```

```
    str1=['Var',num2str(n),' ~ '];
    str2='Var1';
    str3='Var1:Var1';

    for j=2:n-1,
        str2=[str2,'+Var',num2str(j)];
        str3=[str3,'+Var',num2str(j),':Var',num2str(j)];
    end

    gene_modelspec=[str1,'(',str2,')'];

gene_mdl{1,i}=fitglm(gene_ds{1,i},gene_modelspec,'Distribution','bin
omial');
end


% get all p values
all_pvalue=zeros(300,1);
for i=1:300,
    all_pvalue(i,1)=gene_mdl{1,i}.devianceTest.pValue(2,1);
end

% plot using manhattan
manhatten=zeros(300,1);
for i=1:300,
    manhatten(i,1)=-log10(all_pvalue(i,1));
end

plot(manhatten, '.');
```

Problem 3-SKAT 模型:

```
disease <- read.table("ts.txt", header=FALSE)
gene <- read.table("gene.txt", header=FALSE)
#y.b <- as.list(as.data.frame(t(disease)))
y.b <- as.numeric(disease[1,])
Z <- data.matrix(gene)
#Z <- as.list(as.data.frame(t(gene)))
library(SKAT)
obj<-SKAT_Null_Model(y.b ~ 1, out_type="D")
SKAT(Z, obj, kernel = "linear.weighted")$p.value

library(SKAT)
disease <- read.table("txt/disease.txt", header=FALSE)
x1 <- array()
for (i in 1:300) {
    print(i)
    gene <- read.table(paste("txt/gene_",i,".txt",sep=""),header=FALSE)
    y.b <- as.numeric(disease[1,])
    Z <- data.matrix(gene)

    obj<-SKAT_Null_Model(y.b ~ 1, out_type="D")

    x <- SKAT(Z, obj, kernel = "linear.weighted")$p.value
    x1[length(x1)+1]=x
}
```

Problem 3-数据处理：

```
disease = zeros(1,1000);
disease(501:1000) = 1;
dlmwrite('txt\disease.txt',disease,'precision','%d','delimiter', ' ');
load('geno.mat');
j = 0;
for i = 1 : 300
    l = size(importdata(['gene_info\gene_' int2str(i) '.dat']),1);
    dlmwrite(['txt\gene_' int2str(i)
'.txt'],genodata(:,j+1:j+l),'precision','%d','delimiter', ' ')
    j = j + l;
end
```

Problem 4-metaCCA 模型:

```
load('multi_phenos.txt');
load('../geno.mat');
gene = genodata;
fid = fopen('XY.txt', 'w');
fprintf(fid,
'%s%s%s%s%s\n','SNP_id\tallele_0\tallele_1\ttrait1_b\t',...
   'trait1_se\ttrait2_b\ttrait2_se\ttrait3_b\ttrait3_se\t' ,...
   'trait4_b\ttrait4_se\ttrait5_b\ttrait5_se\ttrait6_b' ,...
   'trait6_se   trait7_b    trait7_se   trait8_b    trait8_se' ,...
   'trait9_b    trait9_se   trait10_b   trait10_se');
for i = 1 : size(gene,2)
    fprintf(fid,'%s%d%s','rs', i , ' A T ');
    for j = 1 : size(multi_phenos,2)
        x = gene(:,i);
        y = multi_phenos(:,j);
        lm = fitlm(x,y,'linear');
        fprintf(fid,'%f%s%f%s',lm.Coefficients.Estimate(2),' ',
lm.Coefficients.SE(2),' ');
    end
    fprintf(fid,'\n');
end
fclose(fid);
N1 = 1000;
N2 = 1000;
S_XY1 = importdata('XY.txt');
S_XY1.textdata = [S_XY1.textdata num2cell([zeros(1,20);S_XY1.data])];
t = S_XY1.textdata;
t{1,2} = 'allele_0';
t{1,3} = 'allele_1';
for i = 1 : 10
    t{1,2*i+2} = ['trait' num2str(i) '_b'];
    t{1,2*i+3} = ['trait' num2str(i) '_se'];
end
S_XY1.textdata = t;
S_XY2 = S_XY1;
S_YY1 = estimate_Syy(S_XY1);
S_YY2 = estimate_Syy(S_XY2);
metaCCA_result1     = metaCCA(     2,                      ...
                              S_XY1, S_XY2, ...
                              0, 0,                   ...
                              S_YY1, S_YY2, ...
                              N1, N2 );
```