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		南京邮电大学
参赛队号		10293001
队员姓名	1.	顾凯文
	2.	刘 盼
	3.	汤 健

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：

人体的每条染色体携带一个 DNA 分子，人的遗传密码由人体中的 DNA 携带。全基因组关联性分析（genome-wide association study, GWAS）对大量分析样本建立病例-对照关系，并在全基因组水平上进行分析扫描，最终找出与某一特定表型（疾病）紧密相关的基因标记。GWAS 对于复杂的多基因疾病研究有很好的效果。由于 GWAS 本身的特点，以及大量的数据样本，本文先对样本位点的碱基对编码，通过健康状况或性状关联性分析，发现致病位点以及与疾病相关的基因，主要运用统计学和机器学习算法，分析疾病或相关性状的致病位点与基因，我们做了如下几个方面的工作：

对于问题一，我们综合考虑染色体中 DNA 碱基配对的规则，提出一种数值型的编码方式，作为后续问题的输入，便于数据的统计和分析；

对于问题二，我们分别基于卡方χ^2 检验统计量和随机森林两种算法，提出 M -χ^2 Model 和 M – *Imp Model* 两种问题求解模型，基于这两个模型的检验效果以及准确度检测，提出一种融合前两者优点的“筛选，再筛选，检测”三个步骤的融合模型，得到与疾病相关的 15 个位点为 **rs2273298，rs12145450，rs932372，**

rs2250358，rs9426306，rs4391636， rs12036216，rs4646092，rs7368252，rs7522344，rs2807345，rs11573253，rs15045，rs5746051，rs11580218，将模型所得到的结果，与真实值对比，准确率较前两个模型有所上升，达到 83.4%；

对于问题三，由于基因是由多个位点组成，我们改进χ^2，运用到多个位点的情况，提出一种快速卡方检验的方法，找出与疾病相关的基因，**gene_121，gene_102，gene_125，gene_55 和 gene_62，**；

对于问题四，我们做了相关的数据统计，基于问题二的位点筛选方法和检测方法，把得到的十个性状与疾病相关的位点集做交叉，得出与 10 个性状相关联的位点，为 **rs3218121，rs2273298，rs1553288，rs351617，rs12145450，rs932372，rs2250358，rs12746773，rs12754637，rs4360511，rs12758112，rs1278832，rs35107626，rs716325，rs10737913，rs1775416，rs6577408，rs2526830，rs728340，rs12722898**。

关键词：χ^2检验 遗传统计学 随机森林 全基因组关联性分析(GWAS) 位点(SNPs)

目　录

一 问题重述

近年来，生命科学取得长足进步和分子生物学技术高速发展，生物数据迅速膨胀，越来越来大。为了对这些生物数据进行整理、筛选和分析，生物信息学应运而生。利用计算机技术、统计学与信息学理论和数学方法来解决生物学和生命科学中诸多问题的一门新兴的手段。

染色体是细胞内具有遗传性质的遗传物质深度压缩形成的聚合体。其本质都是脱氧核糖核酸（DNA）和蛋白质的组合（即核蛋白组成的），不均匀地分布于细胞核中 ，是遗传信息（基因）的主要载体。人体的每条染色体携带一个 DNA 分子，人的遗传密码由人体中的 DNA 携带。DNA 的结构很简单，由两条很长的糖链结构构成骨架，通过碱基对结合在一起。在形成稳定螺旋结构的碱基对中共有 4 种不同碱基。分别称之为 A(ADENINE 腺嘌呤)、T(THYMINE 胸腺嘧啶)、C(CYTOSINE 胞嘧啶)、G(GUANINE 鸟嘌呤)。每个基因有几百甚至几万个碱基对。在这条双螺旋的长链中，共有约 30 亿个碱基对，而基因则是 DNA 长链中有遗传效应的一些片段。在组成 DNA 的数量浩瀚的碱基对（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。染色体、基因和位点的结构关系见图 1.

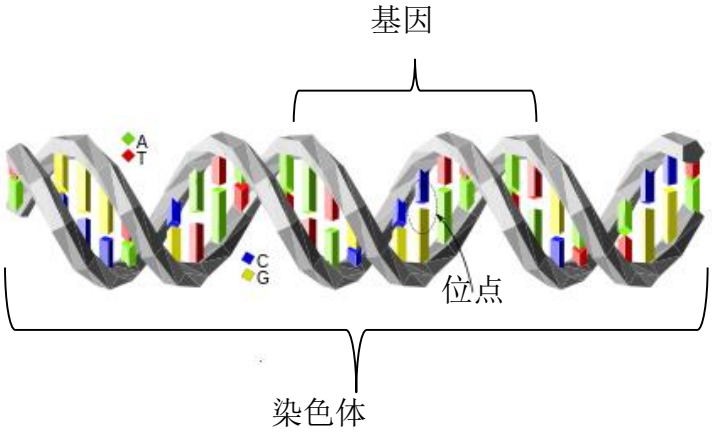

图 1 染色体、基因和位点的结构关系.

在 DNA 长链中，位点个数约为碱基对个数的 1/1000。由于位点在 DNA 长链中出现频繁，多态性丰富，近年来成为人们研究 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。单核苷酸多态性主要是指在基因组水平上由单个核苷酸的变异所引起的 DNA 序列多态性。从分子水平上对单个核苷酸的差异进行检测，SNP 标记可帮助区分两个个体遗传物质的差异。检测 SNP 的最佳方法是 DNA 芯片技术

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相关联，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　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。

本文提出的问题包括：

问题一：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。

二 问题背景及分析

2.1 问题背景

在关联研究方法提出以前，人们主要利用连锁研究方法开展对复杂疾病/性状遗传易感性研究，并且取得一定成绩，发现了一些疾病和性状的易感基因。但由于复杂疾病/性状具有明显的遗传异质性、表型复杂性等特点，使得以家系为基础的“全基因组连锁研究”的应用受到了限制[1]。1996 年，Risch 和 Merikangas 的研究显示在常见复杂疾病的遗传学研究中关联研究较连锁研究有更高的效力，并提出全基因组关联研究（(Genome-wide association study, GWAS）的概念[2]。GWAS 的整体过程比较复杂，其大致流程如下：

(1) 经过处理的 DNA 样品与高通量的 SNP 分型芯片进行杂交；
(2) 通过特定的扫描仪对芯片进行扫描，将每个样品所有的 SNP 分型信息以数字形式储存于计算机中；
(3) 对原始数据进行质控，检测分型样本和位点的得率(call rate)、病例对照的匹配程度、人群结构的分层情况等；
(4) 对经过各种严格质控的数据进行关联分析；
(5) 根据关联分析结果，综合考虑基因功能多方面因素后， 筛选出最有意义的一批 SNP 位点；
(6) 根据需要验证 SNP 的数量选择合适通量的基因分型技术在独立样本中进行验证；
(7) 合并分析 GWAS 两阶段数据。

利用高密度单核苷酸多态(Single nucleotide polymorphism, SNP)标记进行复杂疾病/性状全基因组范围的关联分析已经成为目前分子遗传学领域的研究热点之一[3]。

单核苷酸多态性是指基因组内特定核苷酸位置上存在两种不同的碱基，其中最少一种在群体中的频率不小于 1%。尽管遗传密码由 4 种碱基组成，但 SNP 通常只是 1 种二等位基因的，或二态的遗传变异。SNP 作为一种碱基的替换，大多数为转换，即一种嘧啶碱基换为另一种嘧啶碱基或一种嘌呤碱基换为另一种嘌呤碱基，转换与颠换之比 2：l。SNP 是迄今为止最为丰富精细的遗传标记，人类基因组中共有多少 SNP 位点，目前尚难以确定，这主要是因为还不确知单碱基变异的程度，而各研究者对此估计不完全相同。有人估计每 400bp 就有 1 个碱基不同。一般估计平均每 1000 个碱基就有一个 SNP，在人类 26 条染色体上 30 亿对碱基中，存在 300 万以上的 SNPs。但 SNP 分布并不是均匀的。有根据认为，由于选择压力等原因，SNP 在非转录序列中要多于转录序列、主于基因组中为蛋白质编码的序列仅约为 3%，绝大多数 SNP 位于非编码区。在蛋白质编码区的 SNP 被称为基因编码区 SNP，它们和位于表达调控序列中的 SNP 在功能或致病方面具有更重要的意义[4]。这样的多态性常被称为功能多态性（functional polymorphism）。虽然相对而言，SNP 的多态性不如微卫星多态性（microsatellite polymorphism, MP），但是 SNP 在基因组中数量巨大，分布频密，所以整体而言，其多态性还是比限制性片段长度多态性（restriction fragment length polymorphism, RFLP）与 MP 要高得多。由于每个 SNP 位点通常只含 2 种等位基因（biallele），因此在基因组中，筛查 SNP 往往

只需要进行+/-的分析，而不用分析片段的长度；因而有利于应用大规模自动筛选 SNP 的技术。

随着大数据以及机器学习时代的到来，在本文中，我们提出一种基于统计和机器学习的方法，检测出与某种疾病相关致病位点，和致病基因[5]。

2.2 问题分析

对于问题一，在每个位点的位置，不同样本的编码都是碱基(A, T, C, G)其中两个的组合，则在每个位点位置有三种不同编码方式，如在某一位点是碱基 *T* 和 *C* 的组合，则有 *TT*,*TC* 和 *CC* 组合。分别将如“AA”、“AB” 、“BB”型数据转化为“0”、“1”、“2”，运用这样的数值型编码，简化了后续的分析和计算[6]。

对于问题二，目的是根据患病情况，找出致病位点，与一般的分类问题不同，属于分类问题的逆向分析求解。结合问题一求解的数值编码数据，利用数学统计的卡方检验和机器学习随机森林算法，建立求解模型，找出与疾病相关的一些位点。卡方检验主要基于模型返回的位点的卡方值与显著性水平情况下的卡方值的偏离程度，选取与疾病相关的位点；基于随机森林建立的模型主要根据返回的位点的 importance 的值，并将这些 importance 的值从高到低进行排序，经过网格寻参的方法，取最高的 importance 值的位点，从而找到最有可能致病的位点。鉴于前两个基于卡方检验和随机森林重要性的评分，还提出一种筛选，再筛选，验证的求解步骤。

对于问题三，利用附件的数据，已有 300 个基因样本。基因是有多个位点组成，考虑基因的上位效应，提出一种快速卡方检验的方法，找出最有可能相关的基因；

对于问题四，将 10 个性状看作一个整体，探寻与它们相关的位点。基于此，我们做了相关的数据统计，探索性状的分布以及样本的整体性状分布，基于问题二的位点筛选方法和检测方法，将得到的十个性状与疾病相关的位点集做交叉，得出与 10 个性状相关联的位点。

三 符号说明及名词定义

符号	定义
G	包含 1000 个样本集合
P	是否患病，即表现型数据集
I	300 个基因的数据集
M	附件 multi_phones.txt 的数据集合
S	样本集合
X	位点集合
m	样本数
n	位点数
$A[\,]$	$A[]$保存卡方值大于$\chi_\alpha^2(df)$的位点
$B[\,]$	保存$\chi^2 - Imp\ Model$在迭代 300 次返回位点
$C[\,]$	保存快速卡方检验大于χ_α^2的位点
f_{ij}	χ^2检验中在类别为i，编码为j的观测值
e_{ij}	χ^2检验中的期望频数
M -χ^2	基于χ^2检验返回前 M 个特征模型
M – *Imp Model*	基于随机森林返回前 M 个重要性评分模型
$\chi^2 - Imp\ Model$	基于χ^2和随机森林重要性模型
$Entropy(A)$	属性为 A 的信息熵
$score_i$	位点集合中第 i 个位点的重要性评分
$Gain(A)$	属性 A 的信息增益
Δi	样本不纯度
n_estimators	随机森林树的个数
max_depth	树的最大高度
K-Fold CrossValidatation	K 折交叉验证
$GridSearchCV$	网格搜索交叉验证

注释：$\chi_\alpha^2(df)$为显著性水平下的卡方值

四 问题一求解

4.1 数据编码

对于问题一，DNA 是生物数据库，它的主要功能就是存储包含各种指令的生物信息。DNA 有 G(鸟嘌呤)、T(胸腺嘧啶)、A(腺嘌呤)、C(胞嘧啶)四种碱基，共同构成了相互缠绕的双链阶梯状的螺旋结构。通过这四种碱基不同顺序的编码，存储了生物所有的遗传信息，这样便于对其进行数据分析。将每个位点的碱基(A,T,C,G) 编码方式转化成数值编码方式。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。因此分别将类似于"TT",“TC”,“CC”转化为“0”,“1”,“2”，其他的两个字母组合类似转码。处理这么大的数据量，我们采用 python 语言用 sklearn.preprocessing 包调用 LabelEncoder()函数，将碱基对编码方式转化为“0”,“1”,“2”的数值编码方式。编码之后的碱基对见附件一。

LabelEncoder()将特征标记为 0 到 n_classes - 1 的数，每个位点共有三种编码方式，则 LabelEncoder()可以将每个位点编码为“0”,“1”,“2”三种类型的数值型数据。

编码之前的数据：

表 2 编码之前的位点碱基对

	rs3094315	rs3131972	rs3131969	rs1048488	rs12562034	rs12124819	rs4040617	rs2980300
0	TT	CT	CC	TC	AA	AA	AA	CC
1	TC	CT	CT	TC	GG	AG	AA	CC
2	TT	TT	CC	CC	GG	AG	GG	CC
3	TT	CC	CC	TC	GA	AA	AG	CT
4	TC	CT	CT	TT	GA	AA	AG	CC
5	TC	CC	CC	TC	AA	AA	AA	CC
6	TC	CT	CT	TC	GA	AG	AG	CC

7 rows × 9445 columns

经过编码后的部分数据：

表 3 编码之后的碱基对表示

	rs3094315	rs3131972	rs3131969	rs1048488	rs12562034	rs12124819	rs4040617	rs2980300
0	2	1	0	1	0	0	0	0
1	1	1	1	1	2	1	0	0
2	2	2	0	0	2	1	2	0
3	2	0	0	1	1	0	1	1
4	1	1	1	2	1	0	1	0
5	1	0	0	1	0	0	0	0
6	1	1	1	1	1	1	1	0

7 rows × 9445 columns

五 问题二求解

5.1 基于卡方检验模型（M-χ^2）建立

5.1.1 模型建立

假设：本模型中我们设定的置信度$\alpha = 0.01$。

卡方检验(chi-square test 或χ^2 test）作为非参数检验的一种，是一种常用的对计数资料进行假设检验的统计学方法，主要用于研究 2 组（或多组）样本率或构成比之间的差别，两变量间有无关联性以及频数分布的拟合优度,是用于分类计数资料的假设检验方法，属非参数检验。

对于样本数据，我们使用 spss 工具，对样本的位点 rs3131969 做个初步的统计，统计结果如下图 2 所示，发现该位点的 Pearson 卡方值大于 0.05。

卡方检验

	值	df	渐进 Sig. (双侧)
Pearson 卡方	.290[a]	2	.865
似然比	.291	2	.865
线性和线性组合	.196	1	.658
有效案例中的 N	1000		

a. 0 单元格(.0%)的期望计数少于 5。最小期望计数为 9.00。

图 2 某个位点的 spss 统计

我们建立如下模型，对样本中的位点进行卡方值计算。

卡方检验第一步，建立原假设 H_0 和备择假设 H_1 ；第二步，根据理论经验或理论分布计算期望频数；第三步，根据实际频数和期望频数计算样本卡方

值；第四步，根据自由度和显著性水平 α 在卡方分布表中查找出对应卡方临界值。如果运算出的卡方值大于卡方临界值，接受原假设，反之，接受备择假设。

模型使用列联表卡方检验来检验位点之间差别是否有统计学意义，当待对比样本之间的差异较大时，以此样本为分析对象的卡方计算值就越大。当卡方值大于一定置信水平上的临界卡方值时，则称对比样本之间的差别“有显著性”或“有高度显著性”；反之，卡方值越小，样本之间的差别“无显著性”。检验假设 H_0 为样本间无显著性差异；备择假设H_1为样本间有显著性差异。

由问题一对每一个位点的编码，针对某一个位点S_n我们可以得到如表 4 所示的列联表卡方检验：

表 4 某一个位点S_n的列联表卡方检验

编码 / 类别	0	1	2	
0	f_{00}	f_{01}	f_{02}	$\sum_{j=0}^{2} f_{0j}$
1	f_{10}	f_{11}	f_{12}	$\sum_{j=0}^{2} f_{1j}$
合计	$\sum_{i=0}^{1} f_{i0}$	$\sum_{i=0}^{1} f_{i1}$	$\sum_{i=0}^{1} f_{i2}$	$\sum_{j=0}^{2} f_{0j} + \sum_{j=0}^{2} f_{1j}$

则第n个位点的检验统计量为：

$$\chi_n^2 = \sum_{i=0}^{r-1} \sum_{j=0}^{c-1} \frac{(f_{ij} - e_{ij})^2}{e_{ij}} \tag{1}$$

其中 r 为列联表的行数，表示样本的类别，鉴于数据集只有有患病(1)和不患病（0）两种情况，所以我们取$r = 2$； c 为列联表的列数，表示每个位点有“0”，“1”，“2” 三种编码情况，所以我们取$c = 3$； f_{ij}为观察频数，表示在类别为 i 的情况下，编码为j 的统计的真实值； e_{ij} 为期望频数，可由下面公式计算：

$$e = \frac{RT}{n} * \frac{CT}{n} * n = \frac{RT * CT}{n} \tag{2}$$

RT 为行观察频数的合计；*CT* 为列观察频数的合计；n 为所有观测值的总和，在本模型中可以表示如下：

$$e_{ij} = \frac{\sum_{j=0}^{2} f_{0j} \sum_{i=0}^{1} f_{i0}}{\sum_{j=0}^{2} f_{0j} + \sum_{j=0}^{2} f_{1j}} \tag{3}$$

自由度为$df = (max(i-1)\, max(j-1))$, 则每个位点在卡方检验的过程的自由度为 2。在卡方检验的过程中，我们对位点集合X中的每一个位点计算与其相对应的卡方统计量，即$\chi_n^2 (0 < n \leq 9445, n \in N)$。

在$\chi_n^2 > \chi_\alpha^2(df)$ 的情况下，我们选择接受背择假设H_1，此时说明此位点与样本分类相关性越大。

在基因卡方检验的模型中，由问题一的碱基对数值编码，主要流程如下：

Step1：针对位点集合X中的位点X_n，$(0 < n \leq 9445, n \in N)$，计算$X_n$的卡

方检验量χ_n^2；

Step2：如果$\chi_n^2 > \chi_\alpha^2(df)$，将$\chi_n^2$放入数组$A[\ ]$，$A[\ ]$保存所有有显著性差异的位点的样本的卡方值；

Step3：对$A[\ ]$进行从高到低进行排序，取排序之后的$A[\ ]$前面的k 个位点，作为结果；

Step4：对于选取 k 个位点，进行 10 折交叉验证划分数据集，用随机森林训练得出预测疾病准确率，来检验最优位点的个数。

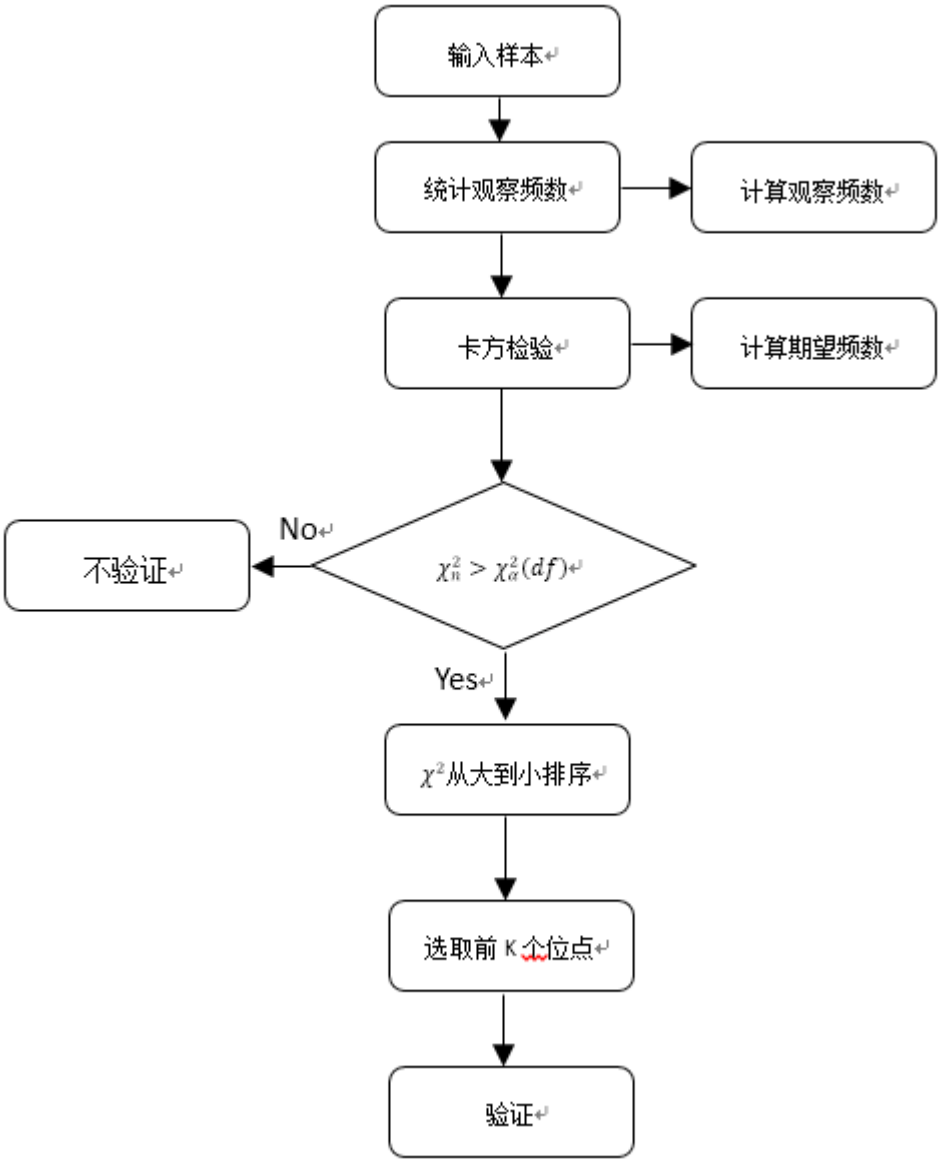

图 3 M-χ^2模型流程图

模型的实现代码见附录一的 *C* 部分。

5.1.2 K-Folds 划分数据集

对于 5.1 模型 Step3 中选取的 k 个位点怎样定义，我们用随机森林预测方法选取的k 个位点的个数对结果准确率的影响变化情况。

K-Folds cross validation（K 折交叉验证法）将待检验数据样本划分为训练样

本和验证样本[7]。K 折交叉验证法一般选用 10 折交叉验证。具体原理用本数据样本解释：将 1000 个样本随机划分到 10 个 folds 内（由于样本标签的正负样本比例 1:1，随机划分到的正负样本基本保持平均），平均每个 fold 有 100 个样本，每次选取其中的 9 个 folds 作为训练样本，另外 1 个作为测试样本。循环 10 次得到全部样本的预测值，与真是标签值比较得到其正确率作为评价选取好坏的标准。

5.1.3 随机森林

在筛选特征时，我们用随机森林求所有特征的重要性排序来筛选特征。在 5.2 节检验部分，划分数据集后我们还用随机森林的分类方法，预测测试集结果，将预测结果与样本的真实值进行比较作为检测结果。

随机森林（Random Forest）由 Leo Breiman 于 2001 年提出，是由多个决策树$\{h(x,\theta_k)\}$组成的分类器,其中$\{\theta_k\}$是相互独立且同分布的随机向量。最终由所有决策树综合决定输入向量 X 的最终类标签[8]。

(1) 决策树

决策树（decision tree）是一个树结构（可以是二叉树或非二叉树）。其每个非叶节点表示一个特征属性上的测试，每个分支代表这个特征属性在某个值域上的输出，而每个叶节点存放一个类别。使用决策树进行决策的过程就是从根节点开始，测试待分类项中相应的特征属性，并按照其值选择输出分支，直到到达叶子节点，将叶子节点存放的类别作为决策结果[9]。ID3 是决策树的经典算法，核心是通过采用信息增益的方式的来选择最好的将样本分类的属性。

设$E = D_1 \times D_2 \times \ldots \times D_n$，是$n$ 维又穷向量空间，其中D_j是有穷离散符号集，E中的元素$e = \{v_1, v_2, \ldots, v_n\}$叫做例子，其中$v_j \in D_j, j = 1,2,3, \ldots, n$。设$s_1, s_2, \ldots, s_n$是$E$ 的 m 个例子集。假设向量空间E中的这 m 个例子集的大小为s_i，决策树基于这样的假设：（1）向量空间 E 上的一棵正确决策树对任意例子的分类概率同 E 中这 m 个例子的概率一致；（2）一棵决策树根据信息熵做出类别选择：

$$Entropy(s_1, s_2, \ldots, s_m) = -\sum_{i=1}^{m} log_2(p_i) \qquad (4)$$

其中$p_i = {}^{s_i}/_{s}$

如果以属性 A 作为决策树的根，A 具有 V 个值，它将 E 分成 V 个子集$\left|E_1, E_2, \ldots, E_V\right|$,假设$E_i$中包含有$S_i(i = 1,2,..,m)$，那么子集 E 所需的期望信息是 E（A）。

$$Entropy(A) = -\frac{\sum_{j=1}^{v}(s_{1j}+s_{2j}+\cdots+s_{mj})}{s} * Entropy(s_1 + s_2 + \cdots + s_m) \qquad (5)$$

因此，以属性 A 为根的信息增益是：

$$Gain(A) = Entropy(A)(s_{1,}s_{2,}\dots,s_{m,}) - Entropy(A) \quad (6)$$

ID3 选择 Gain(A)最大的属性 A*作为根节点，对 A*的不用取值对应的 E 个 V 个子集 E_i 递归调用上述过程生成 A*的子节点，从而生成一棵树。

(2) 随机森林基本原理：

随机森林基本思想是：通过自助法（bootstrap）重采样技术，从原始训练样本集 N 中有放回地重复随机抽取 k 个样本生成新的训练样本集合，然后根据自助样本集生成 k 个分类树组成随机森林，新数据的分类结果按分类树投票多少形成的分数而定。其实质是对决策树算法的一种改进，将多个决策树合并在一起，每棵树的建立依赖于一个独立抽取的样品，森林中的每棵树具有相同的分布，分类误差取决于每一棵树的分类能力和它们之间的相关性。特征选择采用随机的方法去分裂每一个节点，然后比较不同情况下产生的误差。能够检测到的内在估计误差、分类能力和相关性决定选择特征的数目。单棵树的分类能力可能很小，但在随机产生大量的决策树后，一个测试样品可以通过每一棵树的分类结果经统计后选择最可能的分类。

(3) 随机森林的参数调节

应用随机森林方法需要调节几个主要参数：森林中所要生长出的树的个数，生长每棵树中节点分裂随机选择的变量子集中变量的个数，以及每棵树的规模，在用于样本的预测分类的情况下，每个样本所占的权重也可以设置。尽管随机森林方法对参数的设置不太敏感（参数的较大调整获得类似的结果），但是每个参数对偏差和方差的贡献不同，并对最终的结果有一定的影响。

对以上参数的最佳调节要取决于具体的数据集，可以通过理解这些参数是如何影响分类预测的偏差和方差的作用关系，来推测其最佳值，但最终需要实验来决定。本实验的数据维度过大，参数选取范围也较大，参数寻找是个十分费时的过程，手工调参浪费很多时间，我们运用管道 Pipeline 和网络搜索 GridSearchCV 寻参方法寻找最佳参数

Pipeline 能将多个分类器连成一个。当我们用一个固定的顺序处理数据的特征选取、正则化和参数选择，非常有用。这里我们用它来进行分类器参数的网格搜索。GridSearchCV 用来设定每个参数可以搜索到的集合

最终效果最好的参数设定为：

森林树的个数，n_estimators=160；

树的最大深度，max_depth=100；

根据属性划分节点时，每个划分最少的样本数 min_samples_split=7；

叶子节点最少的样本数 min_samples_leaf=10 。

5.1.4 模型检验

对于 5.2 节中划分的测试集和验证集，即对于选取的 k 个位点的样本，使用 K-Folds cross validation 的方法一次划分训练样本和验证样本，再用 Random Forest（随机森林）算法预测样本的值，将预测值与实际样本值比较，判断我们

选取的特征是否是最影响疾病的位点。

表 5 给出选取最相关 k 个位点(属性)与模型准确率之间的关系

表 5 位点数选取与准确率的信息

位点个数 k	模型准确率 p（accuracy）
400	64.01%
600	65.32%
800	69.11%
1000	69.37%
1200	71.42%
1400	72.24%
1600	71.66%
1800	66.32%
2000	63.84%

图 4 给出卡方检验的准确度随着相关位点个数变化的变化趋势。

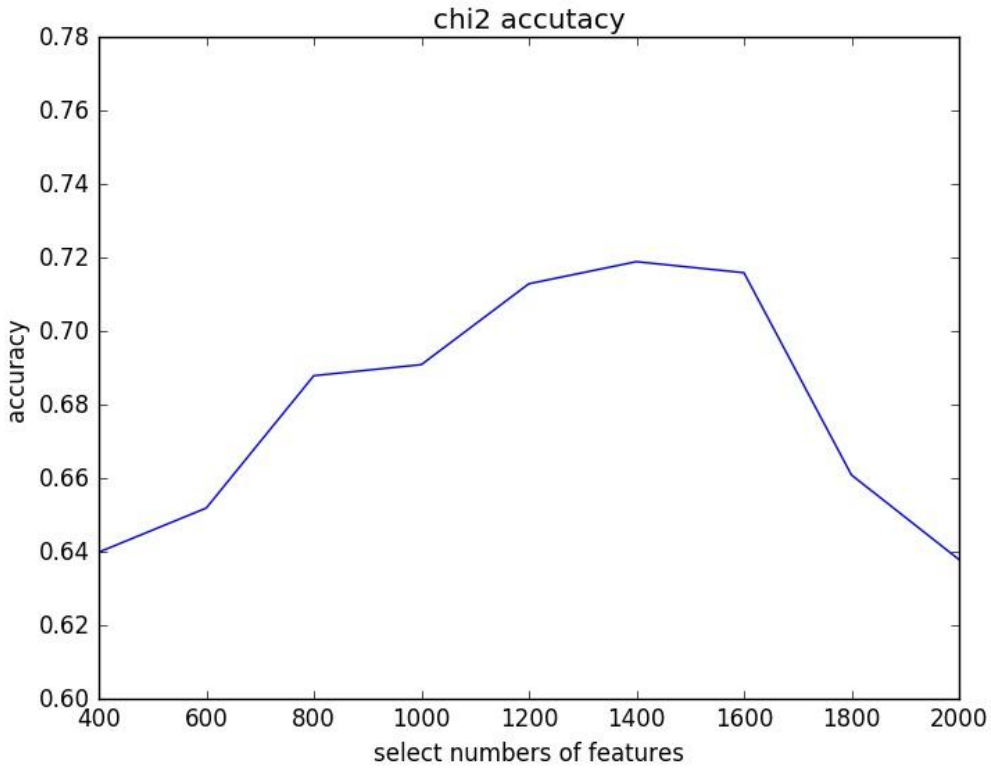

图 4 卡方检验位点个数与准确率的变化情况

由上面的分析可得，在选取前面 1500 个位点时，此模型得到较高的准确率，约为 74.7%，说明我们选的这 1500 个位点对疾病有很大的关系。

5. 2 基于随机森林 Importance 评分模型（*M – Imp Model*）

5.2.1 模型建立

随机森林有一个很大的优点，即在对数据进行分类的同时，可以给出各个基因在分类过程中的重要性 importance 评分，根据该评分能够筛选出相对重要的变量。我们根据 importance 的属性，建立一个基于随机森林—importance 的模型。将随机森林返回的 importance 评分按照从高低的的顺序进行排序，从而可以找到致病位点。

随机森林的基本原理见 5.1.3，我们基于在随机森林分类过程中返回变量的 importance 评分，选取最有可能的致病位点。

(1) 变量重要性值

随机森林方法的一个重要特性是能够计算每个变量的重要性值（VI, Variable Importance ）， RF 提供两种基本的变量重要性值：Gini importance 值和 Permutation importance 值。

(2) Gini importance 值

在节点分裂过程中用 Gini 系数来衡量各节点的样本纯度，Gini 系数定义为：

$$i = 1 - \sum_j p(j)^2 \tag{7}$$

其中，$p(j)$为该节点中属于类别 j 的样本所占的比例。选择合适的节点分裂属性，使子节点的样本纯度比父节点的样本纯度更高，样本的不纯度的下降为：

$$\Delta i = i_{parent} - (p_{left} * i_{left} + p_{right} * i_{right}) \tag{8}$$

p_{left} 和p_{right} 分别为左右两个子节点中样本所占的比例，i_{parent} 、i_{left} 和 i_{right} 分别为父节点和左右子节点的 Gini 系数。任取一个属性变量x_i ，对森林中的所有选择该变量为分裂变量的节点计算不纯度降低量的总和，可获得x_i 的 Gini importance，
即：

$$\Delta \mathrm{I} = \sum_k \Delta i_k \tag{9}$$

(3) Permutation importance 值

通过随机森林 T 中的每棵树 t 对 OOB 样本计算预测准确率A_t。记录 OOB 样本被正确分类的个数为N_r，则：

$$A_t = N_r / N \tag{10}$$

然后将需要计算的变量的属性值打乱随机赋值，再次利用 OOB 样本计算预测准确率A_t^*。记录 OOB 样本被正确分类的个数为N_r'，则：

$$A_t^* = N_r'/N \tag{11}$$

最后将预测准确率的改变量对 T 取算术平均，可获得该变量的 Permutation importance，即：

$$d = \frac{1}{|T|}\sum_{t\in T} A_t - A_t^* \tag{12}$$

样本量为S，表示各样本的变量为各个位点$X_1, X_2, \ldots, X_n (n \leq 9445)$。应用 bootstrap 法有放回地随机抽取 b 个新的自助样本，并由此形成 b 个分类树，每次未被抽到的样本则组成 b 个袋外数据。袋外数据作为测试样本可以用来评估各个变量在分类中的重要性，实现过程如下：

（1）用自助样本形成的每一个树分类器，同时对相应的 OOB 进行分类，得到 b 个自助样本中 OOB 中每一个样品的投票分数，记为$rate_{1,} rate_2 \ldots, rate_b$。

（2）将位点 X_i 的数值在 b 个 OOB 样本中的顺序随机改变，形成新的 OOB 测试样本，然后用已建立的随机森林队新的 OOB 进行分类，根据判别正确的样本数量得到每一个样本的投票分数，所得到的结果可以表示为：

$$\begin{bmatrix} rate_{11} & rate_{12} & \cdots & rate_{1b} \\ rate_{21} & rate_{22} & \cdots & rate_{2b} \\ \vdots & \vdots & \vdots & \vdots \\ rate_{p1} & rate_{p2} & \cdots & rate_{pb} \end{bmatrix}$$

（3）用$rate_{1,} rate_2 \ldots, rate_b$与矩阵对应的第 i 行向量相减，求和平均后再除以标准误差得到位点变量X_i的重要性评分，即

$$score_i = (\sum_{j=1}^{b} \frac{rate_j - rate_{ij}}{b})/S_E, \ (1 \leq i \leq p) \tag{13}$$

由此，我们得到每个位点的重要性评分。即位点X_i 的重要性评分为$score_i$ ，针对每个位点的重要性评分，我们按照从大到小的顺序排列，从中选取一定数量前m 个位点的 importance 值。模型的流程如下：

Step1：对于样本集合S，使用随机森林算法，返回每个属性（位点）的 importance 值，（调用 sklearn 中 RandomForests. feature_importances_函数可以直接返回每个位点的 importance 值），并将这些 importance 值存入数组 *B*[ ]中；

Step2：对于数组 *B*[ ] 中的 importance 值进行从大到小排序；

Step3：选取经过 Step2 排序之后的前 m 个位点；

Step4：依次选取不同的 m 个位点，用随机森林检验，综合选择最合适的 m 个位点。

5.2.2 模型检验

我们用随机森林方法筛选特征，得到重要特征，检验也是运用 5.1.4 节检测方法。按照选取的特征个数和检测结果，分析随机森林方法性能。如下图 4 所示：

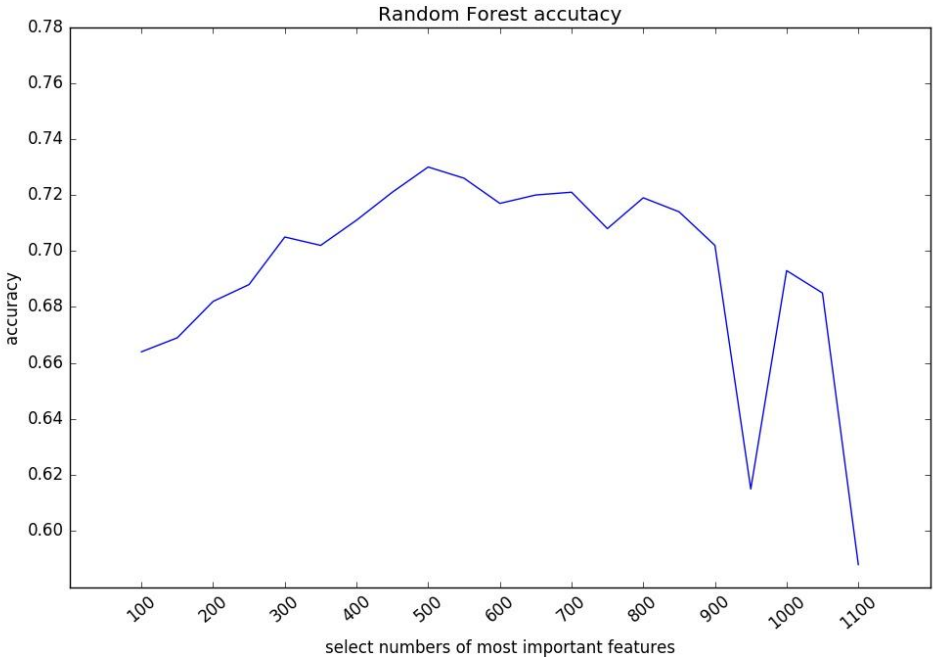

图 5 M-Imp Model 位点个数选择与准确率的变化

观察上图 5 可知，随机森林算法在选取最重要的 500 个位点时，测试效果最好。两种算法的比较观察，卡方检验在筛选到 1500 个特征效果最佳，随机森林在筛选 500 个特征效果最佳。可以得出，随机森林筛选效果更佳，且数据量小时可能随机森林的效果更好。基于两种筛选方法，我们提出了融合两种方法的“筛选-再筛选”的过程。

模型的实现代码见附录 B。

5.3 基于筛选-筛选-验证的模型($\chi^2 - Imp\ Model$)

5.3.1 模型的建立

综合前面两个模型，我们建立一个筛选-筛选-验证的数学模型，主要是基于前面的χ^2检验以及基于随机森林 importance 评分的模型，模型建立的主要流程如下：

Step1：针对样本集合S，先用$M-\chi^2$模型得到相关的 500 致病位点$x_1, x_2, \dots, x_i$；

Step2：针对 Step1 选出的相关位点$x_1, x_2, \dots, x_i$，运用 *M – Imp Model* 模型，返回k 个位点；

Step3：对 Step3，迭代 300 次，记录每次返回的位点，保存在$B[\ ]$；

Step4：对$B[\ ]$中位点进行统计分析，统计每个位点在出现的次数，返回出现频率最高的位点，为最终致病位点。

模型的流程如下图 6 所示：

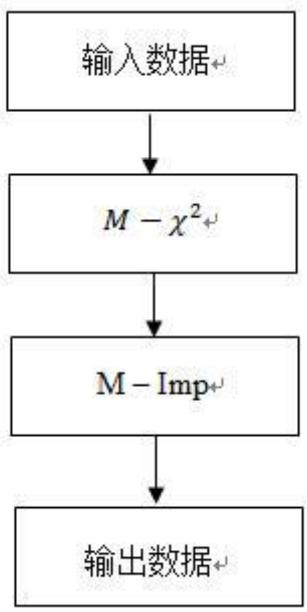

图 6 $\chi^2 - Imp\ Model$流程图

模型的实现代码，见附录 E。

5.3.1 模型验证

根据上述建立的模型，我们针对样本集合S，先经过筛选（χ^2）卡方统计检验，再经过随机森林的重要性评分（*M – Imp Model*）提高了致病位点的准确性。得到以下可能的致病位点：**rs2273298，rs12145450，rs932372，rs2250358，rs9426306，rs4391636， rs12036216， rs4646092， rs7368252， rs7522344， rs2807345，rs11573253，rs15045，rs5746051，rs11580218….**，

具体的位点信息见附录。

表 6 给出相应的位点在 300 此迭代中出现的频数，表中的第一列为致病的位点，第 2 列 Frequency 为相关的位点在 300 此迭代中出现的次数，我们取出现次数最高的位点，为最为相关的致病位点。图 7 给出位点的一个频数的变化情况。

表 6 基于$\chi^2 - Imp\ Model$所得结果的在迭代次数中的频数统计

位点	Frequency
rs2273298	297
rs12145450	231
rs932372	226
rs2250358	218
rs9426306	217
rs4391636	210
rs12036216	195
rs4646092	193
rs7368252	188
rs7522344	173

rs2807345	165
rs11573253	165
rs15045	158
rs5746051	158
rs11580218	152
rs9659647	147
rs7555715	128
rs12133956	128
rs1541318	126
rs2473246	125

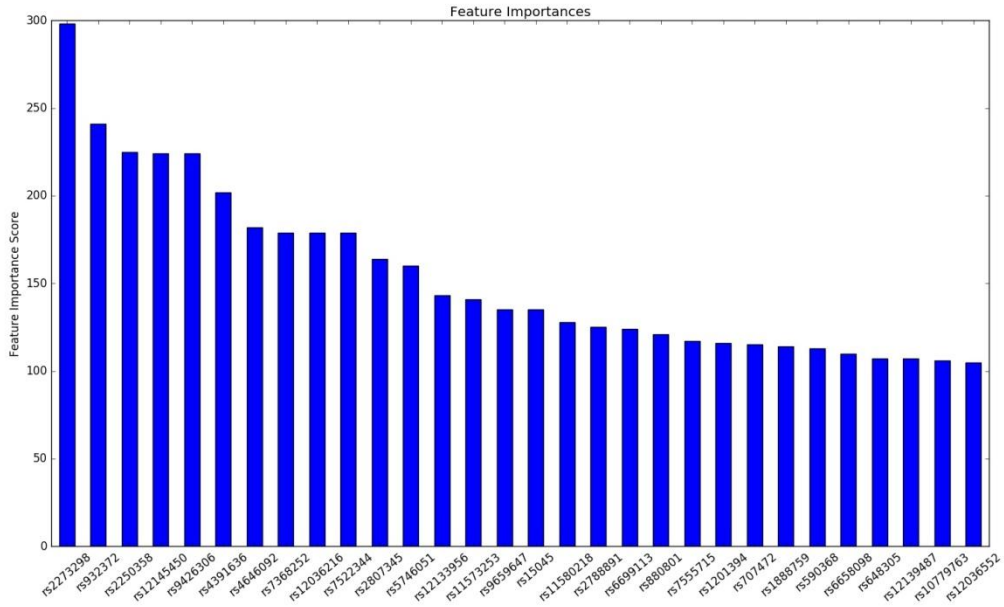

图 7 基于$\chi^2-Imp\ Model$所得结果频数统计

六 问题三求解

6.1 模型建立

基因是若干个位点组成集合，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者其子集合表现出来。

6.1.1 基因的上位效应

在全基因关联分析研究中，随着利用单点测试而找到的一些人类复杂疾病相关的致病基因的发现，人们的兴趣开始转移到对基因-基因或基因环境的交互作用。GWAS 所研究的疾病大致可以分为两类，单基因病和复杂疾病。单基因变异和缺陷是单基因病的主要遗传因素和致病基因。而复杂疾病中，单基因遗传变异难以解释疾病的分子和生物机理，家族连锁分析的作用也非常有限，这一

现象被称为“丢失的遗传性”，即上位效应[10]。大部分的疾病都和很多基因相关联。本题的目的是为了找出与疾病相关的基因，此处需要考虑基因的上位效应。

6.1.2 快速χ^2检验模型

卡方检验是在机器学习和统计领域中属于非参数检验的范畴的一种假设检验方法，在众多领域有着很广的用途，其主要思想是比较样本中实际频数和理论频数的拟合优度或吻合程度问题。

假设本模型中我们设定的置信度$\alpha = 0.01$。

我们有两种类型的数据，表现型数据 P 和基因型数据$I, (I_1, I_2, \dots, I_{300})$，每个基因包含着不同的位点。由统计检验得到，每个基因包含的位点各不相同。由问题一对每个位点的编码，则每一个位点可能的编码为 0,1,2。i表示样本是否患病，$i = 0$表示样本正常，$i = 1$ 表示患病。

我们先定义一个加入某个基因包含两个位点，，令O_{0jk} 表示在此样本在正常样本中，第一个位点的编码为j且第二个位点的编码为 k 的个体的个数，这里 j, k 满足$j, k \in \{0,1,2\}$， 同样的方式我们定义患病的O_{1jk}， O_i表示样本中表现型为 i 的个体的数目，$i \in \{0,1\}$，根据以上定义我们定义以下公式：

$$N_{ijk} = \sum_{i=0}^{1}\sum_{j=0}^{2}\sum_{k=0}^{2} O_{ijk} \tag{14}$$

$$N_{ij} = \sum_{j=0}^{2}\sum_{k=0}^{2} O_{jk} \tag{15}$$

$$N_i = \sum_{i=0}^{1} O_{ijk} \tag{16}$$

则对这两个位点 j,k 的计算的卡方值χ^2为：

$$\chi^2_{jk} = \Sigma_{i=0}^{1}\Sigma_{j=0}^{2}\Sigma_{k=0}^{2}\frac{(O_{ijk}-\frac{N_i N_{jk}}{N_{ijk}})^2}{\frac{N_i N_{jk}}{N_{ijk}}} \tag{17}$$

相应的，假设一个基因包含有多个位点$j, k, \dots, m$ 个位点，则可以基于多维位点建立多维的快速卡方检验[11]。

可以得出多维位点的卡方值为：

$$\chi^2_{jk,..,m} = \sum_{i=0}^{1}\sum_{j=0}^{2}\sum_{k=0}^{2}, \dots, \sum_{m=0}^{2}\frac{\left(O_{(ijk,\dots,m)}-\frac{\Sigma_{i=0}^{1} O_{(j,k,\dots m)}\Sigma_{j=0,\dots,}^{2}\Sigma_{m=0}^{2}(O_{i,j,\dots,m})}{\Sigma_{i=0}^{1}\Sigma_{j=0,\dots,}^{2}\Sigma_{m=0}^{2} O_{(i,j,\dots,m)}}\right)^2}{\frac{\Sigma_{i=0}^{1} O_{(j,k,\dots m)}\Sigma_{j=0,\dots,}^{2}\Sigma_{m=0}^{2}(O_{i,j,\dots,m})}{\Sigma_{i=0}^{1}\Sigma_{j=0,\dots,}^{2}\Sigma_{m=0}^{2} O_{(i,j,\dots,m)}}}$$

(18)

根据以上公式，可以计算每个基因中包含位点的快速卡方检验值。

主要检验流程如下：

Step1: 针对 I 中的每个基因，根据基因的位点个数，结合上述公式，计算关于基因集合 I 的快速χ^2检验值，$\chi^2_{I_1}, \chi^2_{I_2}, \chi^2_{I_3}, \dots, \chi^2_{I_{300}}$；

Step2：针对 Step1 计算出快速χ^2检验，判断这些值是否大于χ_α^2（χ_α^2为显著性水平χ^2检验值，本模型中，假设 $\alpha = 0.01$)

Step3：$C[\,] = [\chi_{I_i}^2, \chi_{I_j}^2, \ldots, \chi_{I_l}^2]$，即$C[\,]$保存与疾病有显著性的基因信息；

Step4：将$C[\,]$中保存的快速检验卡方值进行从高到低排序，选取前 n 个最大的卡方值所对应的基因作为结果。

6.2 模型验证

本模型中，我们运用问题二的随机森林交叉验证模型对本模型进行验证，根据多次验证。

我们选取统计出的前四个基因，作为问题的答案，为 **gene_121，gene_102，gene_125 和 gene_55，gene_62，**作为与疾病最为相关的几个基因，表 7 给出相关基因的卡方值，以及基因中相关位点的信息。

表 7 快速卡方检验模型所得致病基因统计

序号	基因	快速卡方值	相关位点
1	gene_121	29.88	rs590368 ，rs5746051
2	gene_102	28.74	rs2273298
3	gene_125	27.21	rs3013045，rs2999878
4	gene_55	26.50	rs7522344，rs7368252
5	gene_62	20.23	rs2250358
6	gene_265	17.53	rs7543405
7	gene_217	13.2	rs2807345

七 问题四求解

7.1 模型建立

已知 9445 个位点，实际研究把相关性状与疾病看成一个整体，探寻他们和相关位点或基因的相关性。根据疾病相关的 1000 个样本 10 个相关联性信息和其 9445 个位点的编码信息，找出与 10 个性状有关联的位点。

背景：

10 个性状跟遗传疾病有关联，比如高血压，心脏病，脂肪肝和酒精依赖等，分析疾病可以从变现的性状中得到更多有用的信息，提高治病位点或基因的能力。

数据挖掘方法中，如果增加样本的特征，理论上能提高模型的偏差，模型能更好的训练样本，减少欠拟合问题。

这 10 个性状跟遗传疾病有关联，我们把他们分别看成标签特征。第二问题我们分析了位点与疾病的关系，把所有位点看成特征，疾病看成标签，这里我

们类似的分别做了所有位点与 10 个性状的关系，把位点看作特征，10 个性状分别看成标签，分别求出这 10 个标签最相关的位点，将 10 个最相关位点做交集，找到与 10 个性状有关联的位点。

探索数据：

对 multi_phenos.txt 数据导入 python 中，统计探索 10 个性状的样本分布情况，得到如表 8 的数据分布，可以看出每个性状的的样本分布都是平均的，性状表现和不表现样本数都占总样本一半。

表 8 对 multi_phenos 样本的统计分析结果

性状编号	1	2	3	4	5	6	7	8	9	10
性状为 0 样本数	500	500	500	500	500	500	500	500	500	500
性状为 1 样本数	500	500	500	500	500	500	500	500	500	500

我们又对数据样本的 10 个性状做了整体组合，用 hash 函数对每个样本 10 个性状做一次 hash 得到唯一的 hash 值，最后统计 hash 值集合的个数为 271，也就是 1000 个样本一共有 271 种不同性状的人群（从 10 个性状整体来看），说明样本中存在 10 个性状都相同的人群。如果我们将 10 个性状作为整体编码，分析整体与位点的关联性，势必会出现标签类别多，样本数量相对少，而特征多的情况，造成拟合数据的方差较大。而事实上如果这样做，标签类别的分布也不平衡也会造成训练数据之后评价的结果与真实结果差距较大。针对数据探索我们设计了交叉位点法。

数据分析：

Step1: 设计模型找出与性状最相关的位点，

Step2: 迭代 50 回模型，找出频率最高的位点，

Step3: 求与 10 个性状相关的位点的交叉集合。

7.1.2 模型应用分析

为了求解问题四，使用 7.1.1 建立的模型，主要表现在以下几个步骤：

第一步找与性状最相关的位点，我们采用和第二个问题相似的求解方法，并做了模型简化。鉴于随机森林比卡方检验缩小维度能力大，我们选用随机森林的节点分布求位点重要性。随机森林的参数同样用问题二中网格寻参方法。我们设定每次选取前 50 个最重要的特征作为与该性状最相关的特征。

第二步，鉴于第二问随机森林在多次迭代条件下测试下效果优于一次训练结果，我们也对模型迭代了 50 回，取频率最高的特征做为相关特征。这些特征就是与该特性最相关的位点。

第三步，我们得到了 10 个性状相关的位点，用交叉位点法，求 10 个性状的交集。

结果分析得到的结果如表 9 所示，第一列是位点编号，第二列是和位点相关的性状个数。如 rs3218121 位点与 10 个性状的 9 个性状相关，说明该位点与疾病的关联性。我们取关联性状超过 6 个作为该位点与性状关联的标准。

与 10 个性状有关联的位点：**rs3218121，rs2273298，rs1553288，rs351617，rs12145450， rs932372， rs2250358， rs12746773， rs12754637， rs4360511，**

rs12758112，rs1278832，rs35107626，rs716325，rs10737913，rs1775416，rs6577408，rs2526830，rs728340，rs12722898 等 20 个相关的位点。

表 9 求解得到与性状相关的位点及统计

位点编号	相关的性状数
rs3218121	9
rs2273298	8
rs1553288	8
rs351617	7
rs12145450	7
rs932372	6
rs2250358	6
rs12746773	6
rs12754637	6
rs4360511	6
rs12758112	6
rs1278832	6
rs35107626	6
rs716325	6
rs10737913	6
rs1775416	6

八 模型评价与改进建议

8.1 $\chi^2 - Imp\ Model$ 模型的评价与改进

$\chi^2 - Imp\ Model$是在χ^2检验筛选的基础上，经过随机森林的 importance 评分筛选，同时增加迭代的次数，再进行交叉验证，经过双重的筛选，提高了模型的准确率，在迭代一定次数之后，返回在迭代次数中出现频次最高的位点，说明此位点与疾病更为相关，解决了实际性的问题。

此模型在实验的过程中，需要进行多次调试寻参，时间复杂度较高

8.2 快速卡方检验模型的评价与改进

快速卡方检验能够解决多个位点之间和某个疾病之间是否相关联的问题，是χ^2检验是单个位点到多个位点的应用，能考虑到基因之间的上位效应。

位点数量较少时，无法更好的表现多个位点与患病情况之间的是否相互影

响；位点数较多时，会出现观测值数目较多，样本总量数目较少的情况，无法争取反映基因，位点与样本真实的关联性。同时，在改进方面，可以设置一个阈值，对位点数目进行剪枝操作。

8.3 结论

本文所提出的模型，能够解决实际领域的基因与患病是否存在关联，以及找到致病位点和致病的基因，从而更好的定位疾病的症结所在，找到针对性的疾病治疗方案，具有现实性的意义。

九 参考文献

[1] Liu YJ,Liu XG,Wang L,et a1. Genome-wide association scans identified CTNNBLI as a novel gene for obesity. Hum Mol Genet. 2008, 17(12):1803-1813.

[2] 黄文涛，戴甲培，陈润生.全基因组关联研究：进展，问题和未来.中南民族大学学报，2009，28(3)：47-57.

[3] L.Ma,H.B.Runesha et a1.Parallel and serial computing tools for testing single-locus and epistatic SNP effects of quantitative traits in genome-wide association studies. BMC Bioinformatics. 2008, 9:315.

[4] Reich D E, Lander E S. On the allelic spectrum of human disease. Trands Genet, 2007, 17(9): 502-510.

[5] Cook I G, Dummer T J.Changing healh in China:re-evaluating the epidemiological transition model. Health Policy, 2004, 67(3): 329-343.

[6] Keim D A.Information visualization and visual and data mining.IEEE Trasactions on Visualization and Computer Graphics,2002,8(1):1-8.

[7] JiaweiH,Micheline K.Datamining concepts and techniques3rd[M]. America: Morgan Kaufmann, 2012:172.

[8] Hung P.Structure and Light Factor in Different Logged Moist Forests in Huong Son-Vu Quan, Vietnam[M].Cuvillier Verlag,Goetting,Germany.2008.

[9] L. Breiman，Random Forests[J].Machine Learning，2009，45(1):5~32.

[10] 韩建文，张学军. 全基因组关联研究现状[J].2011.1.33(1):25-35.

[11] 周智慧，全基因组关联分析中上位性识别算法的研究及其并行化设计[D],吉林大学.

附 录一

附件说明：文档附件中代码为论文中指示的全部程序代码，可以把每个部分放入用 jupyter notebook 打开的 python 的编辑 cell 中运行。

```
# 问题一
# A 导入数据
import pandas as pd
path = 'C:\Users\Administrator\Desktop\yichuan\genotype.dat'
label_path = 'C:\Users\Administrator\Desktop\yichuan\phenotype.txt'
df = pd.read_table(path,delimiter=' ')                    # 读取样本碱基对
label = pd.read_table(label_path,header=None)             # 读取标签值
df.head(7)

# B 数值编码
from sklearn import preprocessing
le = preprocessing.LabelEncoder()
df_coder = df.apply(lambda x:le.fit_transform(x))         # apply 对每一列编码
df_coder.head(7)

# 问题二
# C 卡方检验筛选
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2                # 导入卡方检验函数
#选择 K 个最好的特征，返回选择特征后的数据
re=SelectKBest(chi2, k=1000).fit_transform(df_coder, label) # 选择 1000 个最好的特征

# D 随机森林
all_imp = []
for i in range(300): # 迭代 300 次
    model_RF = RandomForestClassifier(criterion='entropy',n_estimators=160,
        max_depth=160, min_samples_split=7, min_samples_leaf=10)
    model_RF.fit(df_coder,label)
    important = model_RF.feature_importances_
    imp_index = np.argsort(-important)[:100]              # 筛选前 100 个最重要的特征
    all_imp.extend(imp_index)

# E 统计 300 次迭代后的频率
all_impp = pd.Series(all_imp)
value_count = all_impp.value_counts()                     # 统计频率
value_count_index = value_count[0:100].index              # 频率最高的前 100 个索引
```

 # F 管道和网格搜索给随机森林寻参

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.pipeline import Pipeline
from sklearn.grid_search import GridSearchCV
pipeline = Pipeline([('clf', RandomForestClassifier(criterion='entropy'))])
parameters = {
    'clf__n_estimators': (150,160,170),                    # 设置寻参区间，根据结果不断缩小区间
    'clf__max_depth': (150,160,170),
    'clf__min_samples_split': (6，7，8),
    'clf__min_samples_leaf': (9，10，11)
}
grid_search = GridSearchCV(pipeline, parameters, n_jobs=-1, verbose=1, scoring='f1')
grid_search.fit(de_coder,label[0].values)
print('最佳效果：%0.3f' % grid_search.best_score_)          # 设置评价标准为f1
print('最优参数：')
best_parameters = grid_search.best_estimator_.get_params()  # 返回最优参数值
for param_name in sorted(parameters.keys()):
    print('\t%s: %r' % (param_name, best_parameters[param_name]))
```

 # G 检验（划分+随机森林预测）

```
X=pd.DataFrame(re).as_matrix()
y = np.array(label).T[0]
y_prob = y.copy()
from sklearn.cross_validation import KFold                  # 导入 kfold 函数
kf = KFold(len(y), n_folds=10,shuffle=True)                 # 设置 10 折交叉验证
test_RF = RandomForestClassifier(criterion='entropy',n_estimators=160, max_depth=100,
    min_samples_split=7, min_samples_leaf=9)
for train_index, test_index in kf:                          # 划分训练集和测试集索引
    X_train, X_test = X[train_index], X[test_index]
    y_train, y_test = y[train_index], y[test_index]
    test_RF.fit(X_train,y_train)                            # 训练
    y_prob[test_index] = test_RF.predict(X_test)            # 预测
print np.mean(y==y_prob)                                    # 求准确率
```

 # 第三题

 # H 导入基因数据

```
gene_path = 'C:\Users\Administrator\Desktop\yichuan\gene_info\\'
gene_file_list = os.listdir('C:\Users\Administrator\Desktop\yichuan\gene_info')
gene_dic={}
import os
i=0
for filer in gene_file_list:
    ff = open(gene_path+filer,'r')                          # 循环打开基因位点文件
```

```
        gene_list=[]
        for line in ff.readlines():
            gene_list.append(line.strip('\n'))
        gene_dic[filer] = gene_list                    # 基因位点数据都存储在字典中
        i=i+1
```

I 数据分析

```
from collections import Counter
while i<3:
        while j<3:
            sum01avg=(arr0[i][j]+arr1[i][j])/2
        tmpSum+=pow((arr0[i][j]-sum01avg),2)/sum01avg+pow((arr1[i][j]-sum01avg),2)/sum01av
g
            # print("here is: "+str(i)+" "+str(j))
            j+=1
        i+=1
        j=0
```

 # 第四题

 # I

```
from itertools import combinations
pheno_path = 'C:\Users\Administrator\Desktop\yichuan\multi_phenos.txt'
import pandas as pd
pheno = pd.read_table(pheno_path,delimiter=' ',header=None)
def group_data(data, degree=3, hash=hash, NAMES=None):
        new_data = []; combined_names = []
        m,n = data.shape
        for indicies in combinations(range(n), degree):
            print indicies
            new_data.append([hash(tuple(v)) for v in data[:,indicies]])
            if NAMES != None:
                combined_names.append( '+'.join([NAMES[indicies[i]] for i in range(degree)]) )
        if NAMES != None:
            return (np.array(new_data).T, combined_names)
        return np.array(new_data).T
aa=group_data(pheno.values,10)
```

 # G

```
pheno_path = 'C:\Users\Administrator\Desktop\yichuan\multi_phenos.txt'
import pandas as pd
pheno = pd.read_table(pheno_path,delimiter=' ',header=None)
dic={}
for ii in range(10):
        y = np.array(pheno.ix[:,ii])
        all_imp = []
```

```
for i in range(50):
    model_RF = RandomForestClassifier(criterion='entropy',n_estimators=160,
        max_depth=160, min_samples_split=7, min_samples_leaf=10)
    model_RF.fit(X,y)
    important = model_RF.feature_importances_
    imp_index = np.argsort(-important)[:50]
    all_imp.extend(imp_index)
all_impp = pd.Series(all_imp)
value_count = all_impp.value_counts()
value_count_index = value_count[0:50].index
dic[ii] = df.columns[value_count_index]
print ii
```

附录二

$\chi^2 - Imp\ Model$返回按重要性排序前 500 个位点

rs2273298	rs12139487	rs1569635
rs932372	rs10779763	rs10864301
rs2250358	rs12036552	rs2473345
rs12145450	rs10754873	rs2483274
rs9426306	rs1138333	rs556596
rs4391636	rs1541318	rs10916846
rs4646092	rs7543405	rs262656
rs7368252	rs2473246	rs10916825
rs12036216	rs6429695	rs10915035
rs7522344	rs10779765	rs3762391
rs2807345	rs3013045	rs1193219
rs5746051	rs16830759	rs10910024
rs12133956	rs2038095	rs12097284
rs11573253	rs6683624	rs12128558
rs9659647	rs2982376	rs7553231
rs15045	rs10916703	rs2480772
rs11580218	rs2651935	rs652536
rs2788891	rs9286945	rs4912019
rs6699113	rs2505722	rs1339367
rs880801	rs17356177	rs2821054
rs7555715	rs1257163	rs1009806
rs1201394	rs11249209	rs10492941
rs707472	rs7553424	rs1849943
rs1888759	rs3128342	rs1188399
rs590368	rs761087	rs11584631
rs6658098	rs2143810	rs10864317
rs648305	rs2477777	rs3818033

rs6702295
rs2473247
rs6700387
rs2428556
rs647287
rs351615
rs12126058
rs1883567
rs1891419
rs2797682
rs9729649
rs9430624
rs2092324
rs488595
rs3789543
rs10927586
rs11121742
rs7536195
rs3795438
rs17367504
rs3820514
rs1074078
rs12028945
rs6692372
rs7543064
rs1830705
rs12128253
rs12070592
rs11587046
rs1009113
rs4648537
rs1868302
rs7543486
rs1820205
rs4949238
rs2801178
rs4649002
rs7528781
rs2268170
rs1133398
rs586589
rs6667299
rs6677615
rs17383551
rs1775395
rs1220398
rs6657574
rs7552996
rs4520361
rs2301461
rs4908635
rs11810329
rs4845892
rs2797685
rs679563
rs848214
rs7415936
rs12566535
rs973978
rs12741472
rs7513455
rs2473253
rs8019
rs3766160
rs12144924
rs11590846
rs1033867
rs12077532
rs3010876
rs10927414
rs4662101
rs1203709
rs3795687
rs17356059
rs4908853
rs4661732
rs7518834
rs6661326
rs16851049
rs6661776
rs556258
rs7522712
rs4543765
rs4908748
rs12409315
rs697760
rs2935542
rs12024174
rs473648
rs588641
rs2294811
rs12133334
rs4920381
rs10794531
rs3765695
rs10779722
rs6682378
rs504560
rs905138
rs3890756
rs2898850
rs2270978
rs12752833
rs1292664
rs12760884
rs4520412
rs705579
rs12028120
rs873319
rs6540964
rs10907214
rs2014725
rs12747775
rs12685
rs12402317
rs1695645
rs12736858
rs17458515
rs1005753
rs2236804
rs1924270
rs4543799
rs12117836
rs12139433
rs7539551
rs2182703
rs4846127
rs10803369
rs731024
rs6687869
rs3806310
rs1181876

rs1193223
rs904255
rs3765380
rs1981135
rs11588669
rs4243820
rs4649168
rs2977272
rs7513309
rs11587739
rs28716253
rs4661557
rs11247865
rs10914170
rs6673363
rs946758
rs10864304
rs9329417
rs6696978
rs4845907
rs6678459
rs2473242
rs271383
rs2503009
rs432169
rs34108989
rs873321
rs10462021
rs4466676
rs4073574
rs7533344
rs2272803
rs7540491
rs3002000
rs848203
rs11555809
rs10927972
rs12038524
rs2651927
rs495223
rs11203239
rs11583665
rs1046548
rs10915093
rs3000851
rs6683017
rs12095517
rs2496320
rs11247937
rs4845881
rs4387213
rs6690160
rs16824712
rs1702311
rs10917176
rs17421462
rs9662668
rs12067876
rs1569422
rs10910025
rs7520877
rs11576404
rs4920485
rs12045815
rs598371
rs10489144
rs17184651
rs4908440
rs2227295
rs3010208
rs6697555
rs12747620
rs9661064
rs2744720
rs11121557
rs909948
rs17028511
rs2745302
rs3766306
rs2088824
rs7554327
rs7418164
rs2387698
rs12044299
rs17398063
rs6685177
rs12022529
rs4233292
rs4920460
rs2275819
rs2428735
rs7538084
rs12131096
rs743982
rs11585511
rs1188441
rs1256328
rs2236817
rs4912018
rs226242
rs364642
rs4310409
rs6687987
rs7546786
rs6667416
rs10864479
rs2184708
rs1149046
rs10753253
rs3131419
rs333185
rs3940061
rs12758392
rs2001143
rs10797395
rs12024316
rs6698832
rs2473808
rs11260718
rs4462110
rs6541003
rs10799646
rs675696
rs1065755
rs2789745
rs4080918
rs877648
rs2095518
rs209727
rs2244300
rs10916878
rs6661562

rs12725881
rs7555171
rs2985855
rs516243
rs12022929
rs2748987
rs11573298
rs7526311
rs7528608
rs11590458
rs10492947
rs5022242
rs2071999
rs6426375
rs4654339
rs2359942
rs3738632
rs10928013
rs11261017
rs753305
rs6703610
rs7529591
rs1806990
rs10927632
rs12137409
rs6429674
rs3820610
rs523919
rs2377060
rs1024060
rs3829833
rs1188453
rs7519457
rs377250
rs12078414
rs6688931
rs6681741
rs2242421
rs11121129
rs11260696
rs2097518
rs1763605
rs4912048
rs650298
rs10915317
rs2253372
rs6680132
rs10903032
rs6678862
rs2486669
rs1148455
rs3820609
rs880315
rs914994
rs3010874
rs12128325
rs4649163
rs28603108
rs313990
rs2379159
rs11589934
rs6429732
rs1010082
rs4648464
rs2412150
rs4845812
rs12023288
rs3819967
rs796396
rs17404600
rs17034440
rs7541288
rs7538516
rs198411
rs223183
rs12125512
rs484711
rs4333853
rs12758257
rs6659873
rs2134482
rs10864463
rs17162387
rs6697531
rs2487670
rs4614227
rs12062136
rs517249
rs2981881
rs4600017
rs848195
rs12402628
rs2594289
rs12138909
rs2843404
rs11582200
rs4662000
rs1881561
rs390468
rs502393
rs11576658
rs12567277
rs284272
rs2254669
rs4912122
rs1188394
rs11584888
rs13513
rs11578529
rs12048463
rs848210
rs9426296
rs2020902
rs2293910
rs6429659
rs12021667
rs4661661
rs5064
rs414909
rs1129333
rs1188403
rs4648553
rs6598858
rs1006147
rs6679096
rs2301462
rs665691
rs3003378
rs7551095
rs10465915
rs235219
rs12036784

rs2281303	rs496888	rs10927475
rs4920310	rs7534822	rs12733612
rs7541037	rs12035857	rs1408149
rs4649124	rs4313443	rs6540991
rs6424058	rs11121424	rs10927440
rs7515917	rs12410249	rs551355
rs2445640	rs1148476	rs1193220
rs7541095	rs2820996	