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	西北工业大学
参赛队号	**10699031**
队员姓名	**1.**刘天涯
	2.方　酉
	3.杨冠华

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：

本文建立了基于遗传疾病（性状）与位点、基因之间的关联度的回归模型，实现了单点关联分析、主效基因检测，找出了与某疾病/性状最有可能相关的致病位点和基因。

问题一中，分别使用**三维二值向量**对每个位点的碱基对进行数值编码，如AA=(1,0,0),AT=(0,1,0),TT=(0,0,1)。这种编码方式提取出同一位点处的样本差异信息，忽略具体碱基类型，减小了数据量，通过升维提高了数据提取和分析的灵活性。

问题二属于**单点关联分析**问题。本文结合**多元线性回归**和**参数递减方法**，建立了求解位点与某种疾病之间关联度的**迭代线性回归模型**。首先，分别通过**卡方显著性检验**、**无穷范数**及**相关分析**等方法初步筛选出可能的致病位点，然后使用线性/Logistic 回归模型进行拟合，迭代筛除拟合系数较小的位点以实现降维。拟合过程中采用了**五折交叉验证**方法，将样本分为训练集和测试集，在每次拟合后利用测试集进行验证。最终得到了 5 个最有可能的致病位点：**rs2273298,rs10779765,rs4646092,rs7555715 和 rs7543405**。

问题三属于**主效基因检测**问题，需要整体考虑一个基因内的多个位点。本文在问题二模型的基础上，通过建立基因内回归方程及性状与基因集的回归方程形成了**两层多元线性回归模型**，另外使用了的**最显著SNP**及**Fisher组合法**等方法，最后综合了三种方法的结果，得出结论：与疾病 A 最有可能相关的基因编号为**28、102、150、293**。

问题四需要求得与所给的 10 个性状相关联的位点，即将 10 种性状看为一个整体。本文首先分析了 1000 个样本的性状数量分布情况，发现其中有 294 个样本有所有 10 个性状，300 个样本不带有任何性状，故提取这两个样本集，将 10 个性状降为一维，从而可利用问题二中的模型进行初步求解。之后逐步加入其余样本修正结果。最终找到了**235 个**与 10 个性状有关联的位点。

本文建立的迭代线性回归模型计算量较小、具有一定鲁棒性；求解问题中综合了多种算法的结果，且结果经过了适当的统计检验，证明了模型的正确性。

关键词： 迭代线性回归模型 五折交叉验证 最显著 SNP Fisher 组合法

目 录

一、问题重述

1.1 问题背景

人体的每条染色体携带一个 DNA 分子，人的遗传密码由人体的 DNA 携带。DNA 是由分别带有 A，T，C，G 四种碱基的脱氧核苷酸链组成的双螺旋长链分子。在这条双螺旋的长链中，共有约 30 亿个碱基对，基因是 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（SNPs），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。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。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。

研究人员采用全基因组的方法来确定致病位点或致病基因：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，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。

1.2 问题引入

现有针对某种遗传疾病（简称疾病 A）提供的 1000 个样本信息，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。此外，由于人体的许多遗传疾病和性状是有关联的，可以将相关的形状或疾病放在一起研究（如高，血压、心脏病、脂肪肝和酒精依赖等）因此还提供了上述 1000 个样本的 10 种相关性状的信息，用以补充研究。

需要通过建立数学模型，解决以下几个问题：

问题一：用适当的方法，将样本中的每个位点的碱基(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：根据所提供 1000 个样本在某条有可能致病的染色体片段上的 9445 个位点的编码信息和样本患有遗传疾病 A 的信息，设计和采用一个方法，找出疾病 A 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。现有 300 个基因，每个基因所包含的位点已知。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，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，及其编码信息。在实际的研究中，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。试根据提供的 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，找出与该 10 个性状有关联的位点。

二、模型假设

1. 题中所给数据真实可靠；
2. 给出的位点的碱基编码方式只从数学上考虑其意义而不考虑其生物学意义；
3. 假设致病位点的某个或某些编码会导致个体容易患有疾病 A；
4. 假设各样本之间相互独立。

三、符号说明

符号	符号说明
a_{ij}	健康样本中第 i 个位点第 j 种编码碱基对的总和
b_{ij}	患病样本中第 i 个位点第 j 种编码碱基对的总和
χ_{ij}^2	第 i 个位点第 j 种编码碱基对于患病与否的卡方检验值
$\boldsymbol{X}$	致病相关位点表达组成的向量
$\boldsymbol{Y}$	样本患病情况组成的向量
$\boldsymbol{\beta}$	回归拟合中自变量的系数矩阵
$\boldsymbol{\varepsilon}$	表示随机误差变量的向量
$\boldsymbol{G}$	致病相关基因组成的向量
$x_{i,j}$	第 i 个位点在第 j 个样本中的表达水平
b	系数矩阵中系数元素阈值

四、问题一位点编码方式

4.1 问题描述及分析

原样本信息中所提供的每个位点编码方式为碱基（A，T，C，G）的组合，如位点 rs100015 位置不同样本的编码是 T 和 C 的组合，有三种不同编码方式 TT，TC 和 CC。这样的编码方式使得位点信息清晰，但是不利于下一步数据分析的进行，因此需要将原有编码方式转化成数值编码方式。

对于提供的 1000 个样本，类似地其他位点虽然碱基的组合不同，但不同样本同一位置的位点都是只有两种碱基的组合，即一个位点只能是 A 和 T，A 和 C，A 和 G，T 和 C，T 和 G，或 C 和 G 的组合中的一种，因此只针对于每一个位点所对应组合的种类情况进行编码即可，而每一种碱基组合的只有三种不同编码，即 MM、MN（或 NM）、NN，因此针对每一种组合建立三种数值编码即可。

4.2 数值编码转化

由于数值编码需要体现三种不同编码信息，同时考虑到接下来处理问题的方便性，所以采用 0、1 组合的三位数进行数值编码。具体转化方式如下：

对于 A 和 T 碱基组合的位点，进行如下编码

组合	编码
AA	100
AT/TA	010
TT	001

表 4-1　A、T 碱基组合编码表

对于 A 和 C 碱基组合的位点，进行如下编码

组合	编码
AA	100
AC/CA	010
CC	001

表 4-2　A、C 碱基组合编码表

对于 A 和 G 碱基组合的位点，进行如下编码

组合	编码
AA	100
AG/GA	010
GG	001

表 4-3　A、G 碱基组合编码表

对于 T 和 C 碱基组合的位点，进行如下编码

组合	编码
TT	100
TC/CT	010
CC	001

表 4-4　T、C 碱基组合编码表

对于 T 和 G 碱基组合的位点，进行如下编码

组合	编码
TT	100
TG/GT	010
GG	001

表 4-5　T、G 碱基组合编码表

对于 C 和 G 碱基组合的位点，进行如下编码

组合	编码
CC	100

CG/GC	010
GG	001

表 4-6　C、G 碱基组合编码表

所以对于原样本的位点编码进行转换，对应如下表

样本编号	染色体片段位点名称和对应编码转换						
	rs3094315		rs3131972		…	rs7545865	
1	TT	100	CT	010		GA	010
2	TC	010	CT	010		GG	001
3	TT	100	TT	100		GA	010
4	TT	100	CC	001		GG	001
5	TC	010	CT	010		GA	010
…							
1000	TC	010	CT	010		AA	100

表 4-7 编码转换前后对比表

4.3 编码方式评价

采用新数值编码方式后，列向量的个数由 9445 增加到 3*9445=28335，虽然列向量数目增加了，但是为计算和编程带来了方便。首先将关注点转移到单个位点的碱基组合上，而忽略掉样本位点的横向信息；其次，使用三维向量来表征三种可能的碱基组合，并合理排布，使得数据处理的灵活性提高，在后面的问题处理中使得信息更容易提取，如在提取所有样本某位点某编码的信息时，直接提取该编码数字 1 位所处的列向量即可，使得编程处理更方便，提高效率。

五、问题二模型建立与求解

5.1 问题描述及分析

现有 1000 个样本的信息，其中有 500 个样本患有疾病 A，有 500 个样本未患疾病 A，每个样本信息中都包含了在某条可能致病染色体片段上的 9445 个位点的编码信息。是否患有疾病 A 可能与这条染色体片段上的 9445 个位点中的一个或几个位点相关联，即某个或某几个位点的不同编码形式以及不同编码形式的组合决定了一个个体是否会患 A 疾病。因此为找到 A 疾病最有可能的致病位点，需要找到这 9445 个位点中与样本是否患病相关联显著的位点或位点组合。

由于位点数量较多，直接寻找致病位点较为困难，因此需要采取逐步筛选的方法，寻找与患病与否存在关联的位点。第一级的位点筛选要求较低，可以利用统计中的显著性检验、相关性检验等方法，直接排除掉相关性较差的位点，缩小进一步寻找的范围。再进一步筛选时需要对位点对患病情况的作用进行初步研究，采用多元回归的方法，建立位点与患病情况间的关系式，利用五折交叉验证的方式验证回归式的合理性，同时不断增大回归式中未知数的系数水平，剔除掉系数较小的未知数，从而实现对未知数向量即位点向量的降维。整个问题二的解决过程如图 5-1 所示。

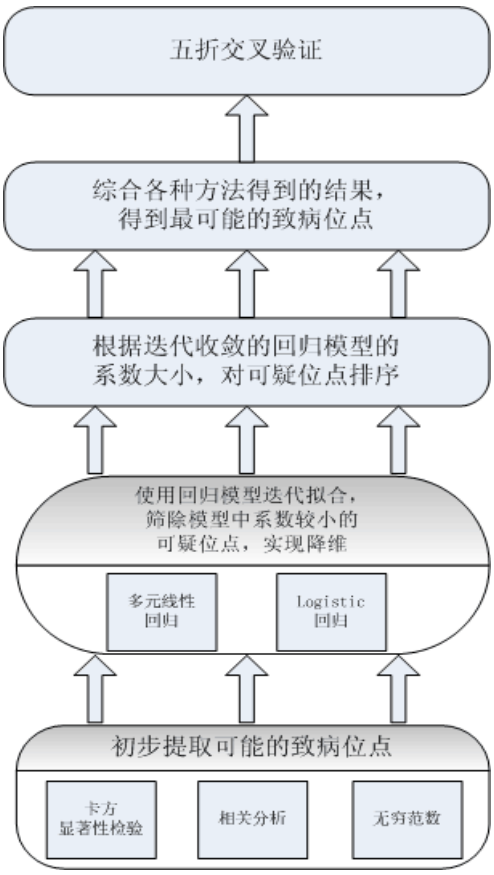


图 5-1 问题二求解流程

5.2 显著性检验

在每一个样本的 9445 个位点中，对于每个位点是否影响样本患病，存在两种可能性：

(1)与样本个体是否患病无关联；

(2)与样本个体是否患病有关联。

由于每个位点都存在三种编码方式，所以对于第(2)种情况与患病相关的位点，又存在两种可能性：

(1)该位点有一种编码方式会导致样本个体患病；

(2)该位点有两种编码方式会导致样本个体患病，也就是说有一种编码方式会使得样本个体不患病。

所以综合以上讨论情况，可以总结：对于每一个位点的每一种编码方式，可能单独存在三种情况：

(1)该位点这一编码方式与是否患病无关，因为该位点与是否患病无关；

(2)该位点这一编码方式会导致个体容易患病；

(3)该位点这一编码方式会导致个体不容易患病；

这样建立了每个位点每种编码方式与个体患病与否的对应，下一步对每个位点每种编码方式与个体是否患病进行显著性检验。

取每一位点每一编码方式与样本个体是否患病进行卡方检验，如对于第一个位点 rs3094315 的各编码方式进行 2X2 卡方检验：

对 rs3094315 位点 TT 编码方式与患病与否进行卡方检验

	健康	患病	总计
TT	a_{11}	b_{11}	$a_{11}+b_{11}$
非 TT	$500-a_{11}$	$500-b_{11}$	$1000-a_{11}-b_{11}$
总计	500	500	1000

表 5-1

$$\chi_{11}^2=\frac{n\left[a_{11}\left(500-b_{11}\right)-b_{11}\left(500-a_{11}\right)\right]^2}{\left(a_{11}+b_{11}\right)\left(100-a_{11}-b_{11}\right)\cdot 500\cdot 500} \tag{5-1}$$

对 rs3094315 位点 TC 编码方式与患病与否进行卡方检验

	健康	患病	总计
TC	a_{12}	b_{12}	$a_{12}+b_{12}$
非 TC	$500-a_{12}$	$500-b_{12}$	$1000-a_{12}-b_{12}$
总计	500	500	1000

表 5-2

$$\chi_{12}^2=\frac{n\left[a_{12}\left(500-b_{12}\right)-b_{12}\left(500-a_{12}\right)\right]^2}{\left(a_{12}+b_{12}\right)\left(100-a_{12}-b_{12}\right)\cdot 500\cdot 500} \tag{5-2}$$

对 rs3094315 位点 CC 编码方式与患病与否进行卡方检验

	健康	患病	总计
CC	a_{13}	b_{13}	$a_{13}+b_{13}$
非 CC	$500-a_{13}$	$500-b_{13}$	$1000-a_{13}-b_{13}$
总计	500	500	1000

表 5-3

$$\chi_{13}^2=\frac{n\left[a_{13}\left(500-b_{13}\right)-b_{13}\left(500-a_{13}\right)\right]^2}{\left(a_{13}+b_{13}\right)\left(100-a_{13}-b_{13}\right)\cdot 500\cdot 500} \tag{5-3}$$

对于剩余其他位点，同样对各位点每一编码方式分别与患病情况进行卡方检验。

对于卡方检验结果，p 值越大，关联效应越弱。针对所有的位点，定义每个位点中 p 值最小的碱基编码的表达水平值为 1，其它两种碱基编码的表达水平值为 0。根据所选用的三位数值编码方式，在相关的计算和编程中，可以提取出该 p 值最小的碱基编码三位数字中为 1 的一列进行运算，这也体现出了第一问所选数值编码方式的优越性。为初步筛除情况(1)中与是否患病无关的位点，给定 p 值阈值 0.01，当某一位点某编码与患病情况卡方检验结果 $p<0.01$ 时，认为有 99%的可信度接受该位点编码与患病与否的相关性，剔除掉不含满足 p 值要求编码的位点；对于含有多个满足 p 值要求编码种类的位点，选择 p 值最低的编码方式，即与患病关联最显著的编码。

经过卡方检验和 p 值阈值条件的筛选后，得到 267 个位点，和它们每一个位点中最可能与样本患病情况相关的编码方式。

5.3 相关性比较

相关性分析是指对两个或多个具备相关性的变量元素进行分析，从而衡量两个变量因素的相关密切程度。因此为获取与是否致病相关的位点，除显著性分析外，还可以进行患病情况与各位点碱基对编码的相关性分析。

具有致病效应的位点在所有样本中的表达水平应与样本类别（即患病和正常）表现出较强的相关性，可以通过位点在样本类别中的表达水平与样本类别间的相关系数作为降维依据，筛选出与类别属性相关性较强的位点。

对于位点 x_i，在样本集中的表达向量 $\boldsymbol{X}_i$ 与样本类别向量 $\boldsymbol{Y}$ 之间的相关系数 $\boldsymbol{R}(\boldsymbol{X}_i,\boldsymbol{Y})$ 计算公式为

$$R(\boldsymbol{X}_i,\boldsymbol{Y})=\frac{\sum_{j=1}^{N}\left(x_{i,j}-\bar{x}_i\right)\left(y_j-\bar{y}\right)}{\sqrt{\sum_{j=1}^{N}\left(x_{i,j}-\bar{x}_i\right)^2\sum_{j=1}^{N}\left(y_j-\bar{y}\right)^2}} \tag{5-4}$$

其中：$N=1000$ 为样本总数；$x_{i,j}$ 为第 i 个位点在第 j 个样本中的表达水平，其取值为 0 或 1，由显著性分析中各位点各种编码对应的 p 值决定，对于 p 值最小的编码取值为 1，其他取值 0；$\bar{x}_i$ 为第 i 个位点在所有样本中的平均表达水平；y_j 为第 j 个样本的类别，取值 0 或 1，0 表示该样本为健康样本，1 表示该样本为患 A 病样本。

由此计算所有位点表达与样本是否患病的相关系数，并对各位点按相关系数由大到小排序，相关系数越接近于 1 则相关性越强。为了与显著性分析所筛选的位点进行匹配对照，选出按照 R 值排序后的前 270 个位点，作为下一步的搜索的范围，这样位点的数据集就从 9445 维降至 270 维。

5.4 逐步降维多元回归

通过显著性检验以及相关性比较得到 267 个或 270 个可能与致病相关的位点以及各个位点最可能与致病相关的碱基编码后，需要进一步研究这些位点的作用。一方面确定各位点对于致病与否的重要性，进而对致病位点进行再次的筛选，从而获取更小范围的可能致病位点；另一方面是研究致病位点间组合作用，可以用作患病与否的检测，以及作为模型优劣检测的标准和工具。由于疾病 A 的致病位点可能不止一个，且可能存在多个位点间的相互作用，所以进行多元回归分析。这里进行迭代的逐步多元回归，考虑两种回归算法，逐步多元线性回归和 Logistic 回归，并对算法效果作比较。

5.3.1 逐步多元线性回归

多元线性回归适用于分析一个因变量和若干个自变量的关系。针对所要研究的关系，可以将该线性回归方程表示为：

$$\boldsymbol{Y}=\boldsymbol{X}\boldsymbol{\beta}+\boldsymbol{\varepsilon} \tag{5-5}$$

这里 $\boldsymbol{Y}$ 表示因变量的向量，即样本患有疾病的情况，患有疾病 A 的元素值为 1，未患疾病 A 的元素值为 0；$\boldsymbol{X}$ 表示所有自变量即初步筛选后的位点的表达值（0 或 1）和一列常数 1 所组成的矩阵；$\boldsymbol{\beta}$ 反映 $\boldsymbol{X}$ 各位点对于患病与否的关联影响程度，$\boldsymbol{\varepsilon}$ 则表示随机误差变量的向量。

为评价回归方程的准确性，采取五折交叉验证法进行测试。在 1000 个样本

随机取出 400 个健康样本和 400 个患 A 病样本，作为训练数据，剩余的 100 个健康样本和 100 个患 A 病样本作为测试数据。将训练数据代入式(5-5)，通过最小二乘法，解得 $\boldsymbol{\beta}$。将剩余的 100 个健康样本和 100 个患病样本测试数据代入所得回归方程，求得结果 $\boldsymbol{y}'$。将结果 $\boldsymbol{y}'$ 中的各元素与实际样本 $\boldsymbol{y}$ 中的各元素一一进行比较，若差值的绝对值 $|y_i' - y_i| < 0.5$，记作一个正确结果，否则记作一个错误结果。定义正确结果数量占测试样本数量 200 的比值为本次回归拟合的正确率，用来评价回归方程的准确性。

对于初步所得 $\boldsymbol{\beta}$，设定阈值 b，将 $|\beta_i| < b$ 的元素，与其所对应的 $\boldsymbol{X}$ 中的位点元素 X_i 从向量 $\boldsymbol{X}$ 中剔除，完成对 $\boldsymbol{X}$ 的降维，得到维度更低 $\boldsymbol{X}'$，然后继续以五折交叉验证的思想，在 1000 个样本中随机抽取 400 个健康样本和 400 个患 A 病样本，作为训练数据，通过最小二乘法解出 $\boldsymbol{\beta}'$，并由剩余样本测试数据计算正确率。若 $\boldsymbol{\beta}'$ 中仍存在 $\left|\beta_i'\right| < b$ 的元素，则继续剔除预期对应的 $\boldsymbol{X}'$ 中的位点元素 X_i'，再次完成对 $\boldsymbol{X}'$ 的降维，得到 $\boldsymbol{X}''$ 并继续进行多元线性回归。以此类推，实现逐步多元线性回归，直到所得 $\boldsymbol{\beta}$ 不再存在 $|\beta_i| < b$ 的元素为止，若此时测试样本正确率仍高于 0.5，则增大阈值 b 的值，继续进行逐步线性回归。直到测试样本的正确率不高于 0.5 时，停止逐步回归过程，即停止 $\boldsymbol{X}$ 的降维过程。

该逐步多元线性回归过程的流程图如图 5-2 所示。

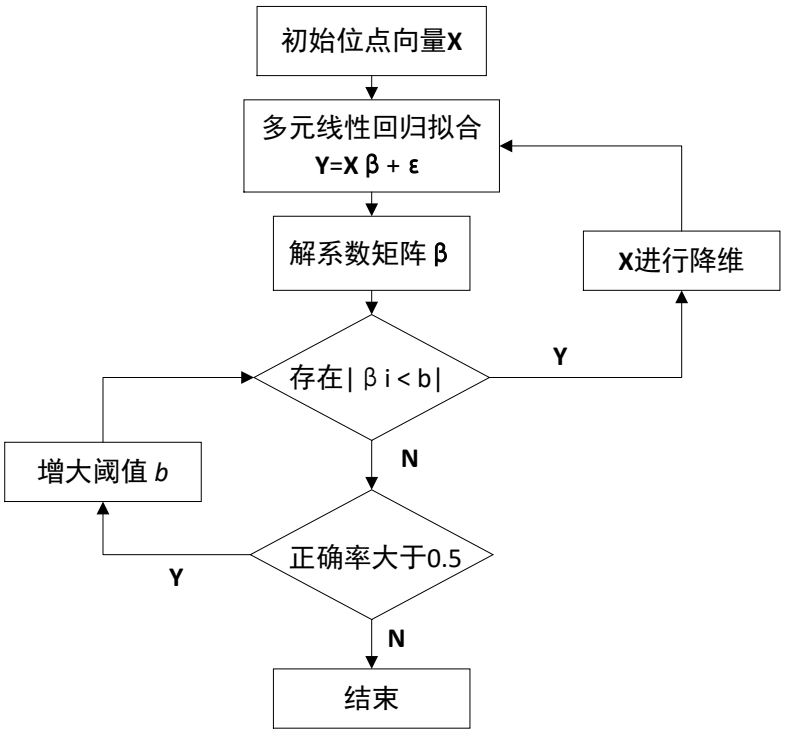


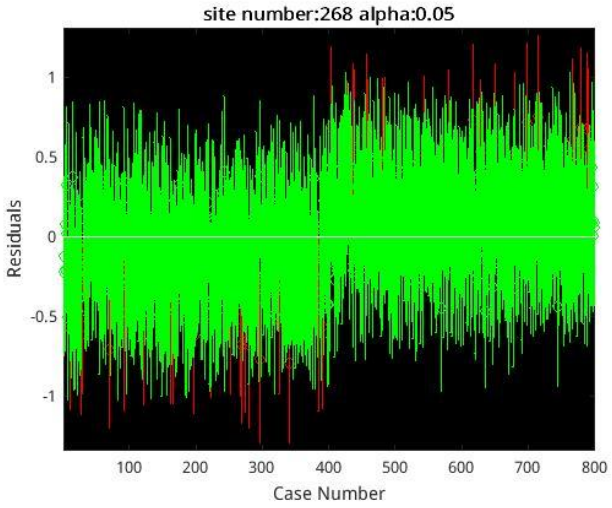
图 5-2　逐步多远线性回归流程图

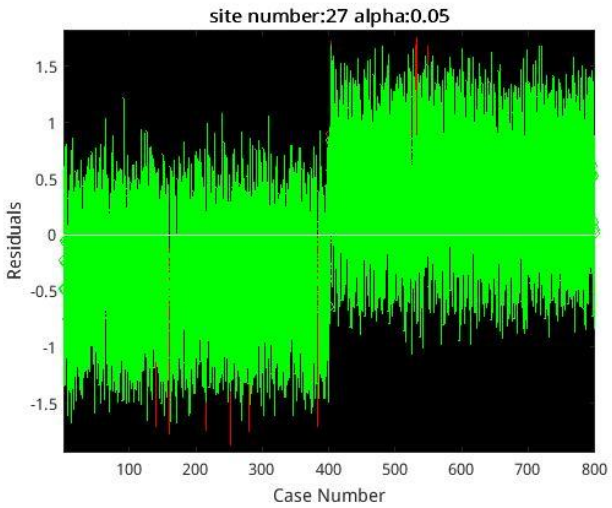
取交叉验证正确率高于 0.5 的最后一次运算的位点向量 $\boldsymbol{X}$，作为逐步多元线性回归实现最大程度降维后的位点集。将每步线性回归的情况列出如表 5-4 所示。

步数	1	2	3
阈值 b	0.01	0.02	0.05
$\boldsymbol{X}$ 维度	268	27	7
正确率	0.81	0.68	0.6

表 5-4　逐步多元线性回归情况

由表 5-4 可知在 $\boldsymbol{X}$ 降维过程中，五折交叉验证的正确率不断降低，也就是说在降维过程中回归的拟合效果不断变差。由各维度所对应的拟合残差杠杆图（图 5-3）中也可看出，残差分布的情况不断变差。





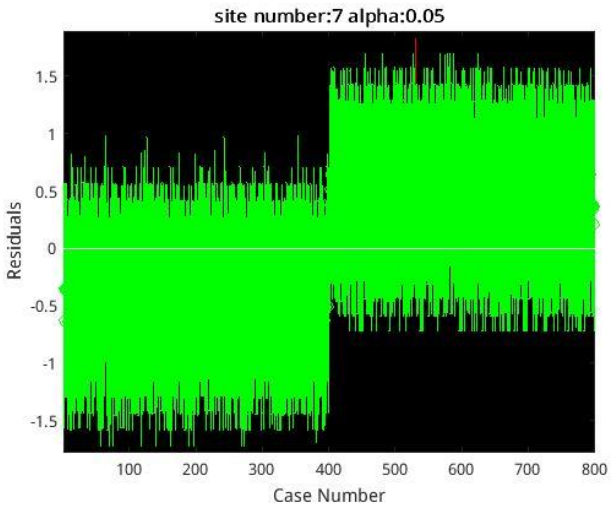


图 5-3　各维度残差杠杆图

5.3.2 逐步 Logistic 回归

Logistic 回归是一种广义的线性回归，其主要适用于因变量为二分类的分类变量。而是否患有 A 疾病是一个二分类的研究目标，满足 logistic 回归的适用条件。通过 logistic 回归分析，可以得到自变量的权重，从而可以大致了解到底哪些位点是可能影响个体患 A 疾病的控制因素。同时根据该权值可以根据位点情况预测一个人患 A 疾病的可能性。

对 1000 个样本的状态（健康=0，患 A 病=1）和 n 个位点之间的关系作 logistic 回归分析，即

$$\mathrm{logit}(p)=\ln\left(\frac{p}{1-p}\right)=\boldsymbol{\beta X} \tag{5-6}$$

与多元线性回归相似，$\boldsymbol{X}$ 表示所有自变量即初步筛选后的位点的表达值（0 或 1）组成的矩阵；$\boldsymbol{\beta}$ 反映 $\boldsymbol{X}$ 各位点对于患病与否的关联影响程度。对模型检验的 p 值可作为位点遗传关联的结果。

与逐步多元线性回归相同，采用五折交叉验证法进行回归分析，根据系数矩阵 $\boldsymbol{\beta}$ 中的元素与阈值的比较完成位点变量矩阵 $\boldsymbol{X}$ 的逐步迭代降维，并根据测试样本的正确率确定最终降维的维度

经过试算，在本例问题中，逐步 logistic 回归拟合结果与逐步多元线性回归拟合的效果几乎没有差别，因此最终选择逐步多元线性回归的方法获取最终降维的位点向量 $\boldsymbol{X}$ 。

5.5 模型求解结果

对通过显著性检验筛取的 267 个位点组成的位点变量矩阵 $\boldsymbol{X}$ 经过逐步多元线性回归降维后，所剩下的位点元素即与致病相关的位点数为 7，分别为第 2938 个位点、第 3341 个位点、第 3398 个位点、第 4526 个位点、第 5944 个位点、第 8380 个位点、第 9196 个位点。

对通过相关性比较筛取的 270 个位点组成的位点变量矩阵 $\boldsymbol{X}$ 经过逐步多元线性回归降维后，所剩下的位点元素即与致病相关的位点数为 9，分别为第 250 个位点、第 1369 个位点、第 2938 个位点、第 3398 个位点、第 4312 个位点、第 4526 个位点、第 5588 个位点、第 8380 个位点、第 9196 个位点。

比较经过初步筛选和多元降维处理后的两组位点集合，取两个位点集合的交集作为最终的结果，可以认定第 2938 个位点、第 3398 个位点、第 4526 个位点、第 8380 个位点和第 9196 个位点为最有可能的 5 个致病位点，即位点 rs2273298、rs10779765、rs4646092、rs7555715、rs7543405。

六、问题三模型建立与求解

6.1 问题描述及分析

基因是 DNA 上具有遗传效应的片段，而每个基因包含一个或多个位点，基因可以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子集表现出来。由于基因对人体性状的表现是通过位点实现的，所以基因对疾病 A 的关联本质上是通过位点的作用实现的，因此研究与疾病最有可能相关的一个或几个基因可以从对位点的研究入手，见微知著。

因此针对本问题，可以建立三种数学模型：多层多元线性回归、最显著 SNP 以及 Fisher 组合法，并根据三种方式的结果进行综合分析得到最终的相关基因。问题三解决流程图如图 6-1 所示。

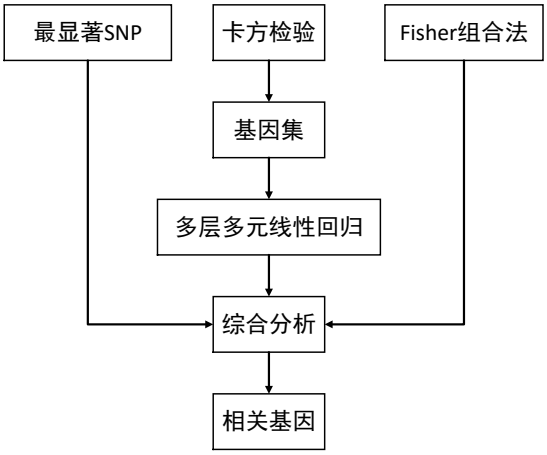


图 6-1 问题三流程图

6.2 逐步嵌套多元线性回归

一个基因含有若干个位点，少则 1 个，多则几十个，且一个基因所包含的不同位点存在绑定关系，即同时存在，且作用系数保持固定比例，基因里的位点具有整体性，因此基因致病的研究与位点致病的研究不能混为一谈，但是两者的研究又存在共性。

同问题二求解，同样对样本所提供的所有位点与样本是否患病进行显著性分析，根据 $p<0.01$ 的要求对位点进行初步筛选，得到 267 个相关程度高的位点组成向量 $\boldsymbol{X}$ 。由于每个位点都属于且只属于一个基因，因此向量 $\boldsymbol{X}$ 映射了包含它们的基因组成的向量 $\boldsymbol{G}$ ， $\boldsymbol{G}$ 的元素数量不超过 $\boldsymbol{X}$ 。

针对所要研究的基因与患病情况的关系，建立线性回归方程表示为：

$$\boldsymbol{Y}=\boldsymbol{G}\boldsymbol{\alpha}+\boldsymbol{\varepsilon} \tag{6-1}$$

同时，基因的表达取决于所包含的位点，因此

$$G_i=\boldsymbol{X}_i\boldsymbol{\beta}_i \tag{6-2}$$

其中G_i表示基因组成向量$\boldsymbol{G}$中的第 i 个元素，也就是第 i 个基因；$\boldsymbol{X}_i$为第 i 个基因所包含的位点集合向量，而$\boldsymbol{\beta}_i$为第 i 个基因中所包含的所有位点对于该基因的表达系数水平，两式形成嵌套的线性回归方程。

对此回归方程的处理思想类似于问题二，即由初值开始，进行多元线性回归拟合，并通过系数向量$\boldsymbol{\alpha}$元素与阈值的比较，实现基因向量$\boldsymbol{G}$的降维。对于降维过程中回归方程拟合情况的评价，同样采取五折交叉验证的方法，并保证样本测试正确率高于 0.5。

而与第二问不同的是，这里还存在基因对位点的包含嵌套问题，因此需要进行多一步运算。

经过逐步迭代求解，最终得到最有可能与疾病相关的几个基因编号是 28、50、102、150、193、293。

6.3 最显著 SNP

基因通过位点的作用对疾病性状产生作用，因此与是否患病关联性较强的位点同时会使得该基因与是否患病较强关联，因此可以用该基因中 p 值最小的位点代表该基因，得到样本中 300 个基因的代表 p 值，将基因按照 p 值从小到大排序，并提取出排序在前十位的基因，即为最有可能与疾病 A 相关的基因。所提取基因与其对应 p 值如表 6-1 中所示。

基因编号	基因 p 值
102	1.24554e-05
115	0.000146166
150	0.000147915
114	0.000338403
193	0.00048475
293	0.000499405
265	0.000659608
298	0.000723028
28	0.000943092
254	0.00127404

表 6-1　最可能致病基因与对应 p 值

6.4 Fisher 组合法

假设基因上的所有位点都不相关，所有的 p 值相互独立，且分布均匀，则可以利用组合的方法将全部的位点的遗传关联结果联合起来形成全部基因的遗传

关联结果。Fisher 组合法是所有组合法的基础，是一种基因关联分析方法。

对样本中每一个基因上的所有位点的 p 值进行直接乘积，并检验统计量

$$X = -2\ln\left(\prod_{i=1}^{n} p_i\right) = \sum_{i=1}^{n} -2\ln\left(p_i\right) \tag{6-3}$$

服从自由度为 2n 的 χ^2 分布。

取满足 p<0.05 的结果，所得到的基因结果如表 6-2 所示。

基因编号	p 值	基因编号	p 值
293	0	149	0.025685
115	7.06637e-06	229	0.0257481
150	5.40369e-05	235	0.027375
78	0.000145506	42	0.0290914
102	0.000328547	103	0.0296084
217	0.000461643	297	0.0304095
153	0.00163836	194	0.0333076
55	0.00178332	192	0.0344978
22	0.00259503	132	0.0359497
169	0.00472419	265	0.0367117
121	0.00639707	30	0.0394734
279	0.00913731	90	0.0462222
9	0.00917955		
28	0.0108978		
45	0.0146495		

表 6-2 满足 p<0.05 的基因点与其 p 值

6.5 模型求解结果

最显著 SNP 与 Fisher 组合法是目前进行全基因组关联性分析时常采用的方法。将通过逐步嵌套多元线性回归拟合最终得到的结果与这两种常用方法结果作比较，可以看出逐步嵌套多元线性回归拟合具有一定的普适性。进一步通过三种模型的结果比较分析，与疾病 A 最有可能相关的基因编号为 28、102、150、293。

七、问题四模型建立与求解

7.1 问题描述及分析

人体的许多遗传疾病和性状是有关联的，因此可以把许多相关的性状或疾病放在一起研究，从而提高发现致病位点或基因的能力。现有 1000 个样本的 10 种相关性状信息，以及样本的位点编码信息，需要获取与此 10 个性状相关联的位点。由于这 10 个性状是相关联的，所以极有可能出现显示全部 10 个性状的个体样本以及 10 个性状全部都不显示的个体样本。通过对样本的初步统计发现，这 1000 个样本中显示全部 10 个性状的个体有 294 个，10 个性状全部都不显示的个体有 300 个。这两类性状非常明显的个体总数大约占总样本的 60%，因此具有

较高代表性，可以对这 594 个个体进行性状位点关联研究。

7.2 相关性分析

由于这 594 个样本的性状都较为统一，可以对每个样本的性状向量降维，用 1 和 0 分别表示显示性状和不显示性状,这样就把本题简化成了问题二中的问题。采用问题二中相关性比较，计算各位点与性状是否显示的相关系数，并选取相关系数 R 阈值为 0.5，筛选出相关系数绝对值大于阈值的位点，即为与这 10 个性状有关联的位点。

7.3 模型求解结果

经过筛选，得到与这 10 个性状有关联的位点序号为 59、 63、 64 、67、 74、 189、 223、 225、 227、 263、 273、 527 、604、 616、 618、 699、 715、 1005、 1050、 1116、 1121 、1167、 1182、 1202、 1257、 1339、 1369、 1494、 1585、 1600 、1629、 1656、 1777、 1884、 1887、 1907、 2020、 2046、 2049 、2050、 2054、 2123、 2138、 2155、 2299、 2451、 2488、 2521 、2535、 2587、 2708、 2770、 2806、 2821、 2832、 2848、 2866 、2883、 2908、 2982、 2998、 3030、 3121、 3134、 3175、 3299 、3345、 3376、 3456、 3770、 3829、 3853、 3887、 3899、 3922 、3927、 3929、 3931、 3945、 4046、 4051、 4086、 4111、 4335 、4467、 4475、 4477、 4499、 4527、 4578、 4632、 4634、 4682 、4684、 4710、 4749、 4757、 4770、 4775、 4777、 4786、 4791 、4792、 4793、 4795、 4801、 4919、 4966、 5065、 5068、 5092 、5141、 5153、 5172、 5209、 5279、 5359、 5427、 5594、 5624 、5664、 5691、 5722、 5737、 5803、 5836、 5876、 5887、 5893 、5897、 5898、 5963、 6041、 6048、 6064、 6073、 6181、 6215 、6223、 6272、 6296、 6317、 6360、 6367、 6376、 6403、 6450 、6488、 6564、 6650、 6716、 6722、 6741、 6816、 6951、 6986 、7030、 7056、 7061、 7105、 7181、 7212、 7257、 7280、 7282 、7287、 7342、 7348、 7392、 7450、 7471、 7494、 7538、 7575 、7648、 7655、 7688、 7798、 7904、 8000、 8036、 8038、 8045 、8062、 8064、 8078、 8113、 8226、 8298、 8329、 8337、 8343 、8346、 8360、 8363、 8368、 8387、 8420、 8430、 8456、 8469 、8484、 8489、 8535、 8551、 8557、 8576、 8631、 8646、 8690 、8691、 8693、 8699、 8709、 8756、 8803、 8804、 8808、 8811 、8821、 8822、 8896、 8942、 9026、 9040、 9097、 9234、 9243 、9250、 9308、 9358、 9374、 9381、 9408 共 235 个。

八、模型评价与展望

8.1 模型评价

优点

1. 问题一中设计的编码方式提取出了位点中的关键信息，剔除了判断致病位点不需要的横向信息（即具体的碱基组合），减小了数据量。使用三维向量来表征三种可能的碱基组合，提高了数据处理的灵活性，更容易提取显著性强的位点；
2. 问题二、三建立的迭代降维多元线性回归模型使用了参数递减方法，由较多的可疑位点出发，逐步降维，最终获得较少数量的最可能的致病位点（基因），具有较好的鲁棒性；
3. 提取致病位点（基因）时都使用了两到三种独立的算法，最终经过综合比对确定最可能的致病位点（基因），各种算法互为验证，提高了结果的正确性；
4. 将样本分为训练样本集和测试样本集，使用五折交叉验证对模型进行了检验，证明了模型的合理性；
5. 使用的多元线性回归等算法计算量较小，可推广应用到更大的数据集。

缺点

1. 问题二中只对单个位点进行了分析，而未进行多个位点的关联分析。这样就遗漏了与疾病存在关联但关联效应较弱的致病位点，造成判断结果的不准确；
2. 在降维过程中，待拟合参数逐渐减少，造成样本数量远大于参数数量，易出现过拟合现象，拟合效果变差；
3. 重点分析了病人与健康者之间的位点差异，而忽视了病人、健康者各自集合内的位点分布特征，可能遗漏了有用信息。

8.2 改进方向

接下来需要在以下几个方面对模型进行改进：

1. 增加单点关联分析，如使用非线性回归模型进行拟合，挖掘多个位点的共同作用；
2. 使用岭回归解决多重共线性问题；
3. 根据待拟合参数数量优化训练样本集规模，改善过拟合现象。

九、参考文献

[1] 罗旭红, 刘志芳, 董长征. 基因水平的关联分析方法[J]. 遗传, 2013, 35(9):1065-1071.
[2] 黄文涛, 戴甲培, 陈润生. 复杂疾病全基因组关联研究:进展,问题和未来[J]. 中南民族大学学报(自然科学版), 2009, 28(3):47-57.
[3] 刘全金, 李颖新, 阮晓钢. 基于统计方法的肿瘤特征基因提取[J]. 北京工业大学学报, 2005, 31(2):122-125.
[4] 陈峰, 柏建岭, 赵杨,等. 全基因组关联研究中的统计分析方法[J]. 中华流行病学杂志, 2011, 32(4):400-404.
[5] 杨昭庆, 洪坤学. 单核苷酸多态性的研究进展[J]. 国际遗传学杂志, 2000(1):4-8.
[6] 陈鸿建. 概率论与数理统计[M]. 高等教育出版社, 2009.
[7] 谢宇. 回归分析[M]. 社会科学文献出版社, 2013.
[8] 郝柏林. 来自基因组的一些数学[M]. 上海科技教育出版社,2015.

十、附录

multiple_linear_regression

```
%%
% Configurations.
% Choose encoding mode, 1 for 3 bits; 0 for 0~2.
use_genotype_3x = 1;
% Choose multiple linear regression method.
% 1: regress; 2: stepwisefit; 3: robustfit(logistic).
% TODO: REMOVE 2!!!
reg_method = 1;
% Method used to extract possible pathogenic sites(bits).
% 1: chi-square test; 2: infinite norm.
p2_extract_method = 1;
% Save result to .mat file?
p2_save_result = 0;
% Iterate regression to reduce dimension of final result?
iterate = 1;
% Terms with oefficients smaller than 'min_coef' will be removed
% in the iteration to reduce dimension of final result.
% try: 0.05; 0.1; 0.15; 0.2.
% (0.2 is already too much. Dimension is reduced to 4,
%    elements in Yhat(Y_test) are almost same.)
min_coef = 0.05;
%-------------------------------------------------------------
% Threshold(p) in chi-square test.
% Takes value in [0.01;0.001;0.0001]
threshold = 0.01;
%-------------------------------------------------------------
% Amount of bits extracted using infinite norm.
% Takes value in [1000;300;200;100;50]
amount_of_bits_extracted = 300;
%-------------------------------------------------------------
%%
% Load possible_pathogenic_idx from .mat file.
if p2_extract_method == 1      % chi-square test.
    str1 = 'chi2';
    switch threshold
        case 0.01
            str2 = 'p_0_01';      % extract 276 sites.
        case 0.001
            str2 = 'p_0_001';     % extract 24 sites.
        case 0.0001
```

```
                str2 = 'p_0_0001';      % extract 5 sites.
            case 0.05
                str2 = 'p_0_05';
        end
%-------------------------------------------------------------
elseif p2_extract_method == 2      % infinite norm.
        str1 = 'inf_norm';
        str2 = num2str(amount_of_bits_extracted);
else
        disp('p2_extract_method is invalid!')
end

mat_name_str = ['p2_' str1 '_pathogenic_idx_3x_' str2 '.mat'];

% ALWAYS load data from file!!!
% if ~exist('possible_pathogenic_idx','var')
        load(mat_name_str)
        fprintf('Loading from %s...\n',mat_name_str)
% end

clear p2_extract_method mat_name_str

%%
switch reg_method
        case 1
                reg = 1; stepwise_reg = 0; logistic = 0;
        case 2
                reg = 0; stepwise_reg = 1; logistic = 0;
        case 3
                reg = 0; stepwise_reg = 0; logistic = 1;
end

%%
% Multiple linear regression analysis on the
% possible pathogenic sites(bits) obtained above.
%-------------------------------------------------------------
% Solve for the vector B of regression coefficients
% in the linear model Y = X * B.
% Y: phenotype;
% X: most distinct n columns in genotype_3x corresponding to each of the
%               possible pathogenic sites obtained above.
Y = phenotype;
% Remove 200 testing samples. Cannot change order of next 2 lines
% because it causes 'Matrix index is out of range for deletion.'
```

```
Y(901:1000,:) = [];
Y(401:500,:) = [];
X = [];

if use_genotype_3x
%-------------------------------------------------------------
for i = 1 : length(possible_pathogenic_idx)
    X = [X, genotype_3x(:,possible_pathogenic_idx(i))];
end
%-------------------------------------------------------------
else % use genotype(0~2)
%-------------------------------------------------------------
for i = 1 : length(possible_pathogenic_idx)
    X = [X, genotype(:,possible_pathogenic_idx(i))];
end
%-------------------------------------------------------------
end
% Remove 200 testing samples.
X(901:1000,:) = [];
X(401:500,:) = [];

%%
%==============================================
% Prepare test samples 'X_test' for calculating model accuracy
% later in 'p2_s3_calc_correct_pct.m'.
% This part is moved from 'p2_s3_calc_correct_pct.m'.
% Very UGLY but it works...
% TODO: figure out how to deal with this problem in p2_s3.
X_test = [genotype_3x(401:500,possible_pathogenic_idx); ...
            genotype_3x(901:1000,possible_pathogenic_idx)];

% Add constant term to X_test if using regress() or robustfit()!!!
if reg || logistic
    X_test = [ones(200,1),X_test];
end
%==============================================

%%
if reg
%-------------------------------------------------------------
% B = regress(Y,X) returns the vector B of regression coefficients
% in the linear model Y = X * B.
% NOTICE: must add constant term to the left of X!
X = [ones(num_samples - 200,1),X];      % add constant term to X.
```

```
alpha = 0.05; % def: 0.05
[B,BINT,R,RINT,STATS] = regress(Y,X,alpha);

% STATS

% STATS contains the R-square statistic, the F statistic and p value
% for the full model, and an estimate of the error variance.
if STATS(1) >= 0.95          % R-square > 0.95
    fprintf('Multiple linear fitting succeeded!\n');
end

Yhat = X * B;
disp('Regress fit finished.')
fprintf('Originally X has %d columns.\n',size(X,2))

%  残差与残差区间的杠杆图
rcoplot(R,RINT)
str_title = ['site number:' num2str(size(X,2)) ...
    ' alpha:' num2str(alpha)];
title(str_title)
%-------------------------------------------------------------
if iterate
%-----------------------------------------
% Already added constant term above!
% X = [ones(num_samples - 200,1) X];

% When threshold = 0.01:
% (筛除系数小于 0.05 的项，可降维至 111.)
% Iterate to remove terms with coefficients < 0.05, B dimension -> 111;
% Iterate to remove terms with coefficients <    0.1, B dimension ->    31;
% Iterate to remove terms with coefficients < 0.15, B dimension ->      8;
% Iterate to remove terms with coefficients <    0.2, B dimension ->      4.
last_B_dim = 0;

fprintf('Start iteration to reduce dimension, min_coef = %g:\n',min_coef)
while 1
    for j = length(B) : -1 : 2 % Ignore column 1(constant term).
        if abs(B(j)) < min_coef
            % Remove terms with coef less than min_coef.
            X(:,j) = [];
            % Remove corresponding columns in X_test.
            X_test(:,j) = [];
        end
    end
```

```
        fprintf('Now X has %d columns.\n',size(X,2)) % DEBUG
        [B,BINT,R,RINT,STATS] = regress(Y,X,alpha);
        Yhat = X * B;

        % break condition: the dimension of stops decreasing.
        if last_B_dim == length(B)
            break;
        end
        last_B_dim = length(B);
end
%  残差与残差区间的杠杆图
figure(2)
rcoplot(R,RINT)
str_title = ['After iteration. site number:' num2str(size(X,2)) ...
        ' alpha:' num2str(alpha)];
title(str_title)
disp('Iteration finished.')
clear last_B_dim
%----------------------------------------
end
%------------------------------------------------------------
end
%%
% stepwise regression
if stepwise_reg
%------------------------------------------------------------
PENTER = 0.05;
PREMOVE = 0.10;
[B,SE,PVAL,INMODEL,STATS,NEXTSTEP,HISTORY]                                    =
stepwisefit(X,Y,'penter',PENTER,'premove',PREMOVE);

Yhat = X * B;
disp('Stepwise fit finished.')
clear SE PVAL INMODEL NEXTSTEP HISTORY % Don't need them now.
%------------------------------------------------------------
end

%%
if logistic
%------------------------------------------------------------
[B,STATS] = robustfit(X,Y,'logistic');
Yhat = [ones(num_samples - 200,1) X] * B;
disp('Logistic fit finished.')
fprintf('Originally X has %d columns.\n',size(X,2))
```

```
if iterate
%----------------------------------------
% Add constant term to the left of X.
X = [ones(num_samples - 200,1),X];
% When threshold = 0.01:
% (筛除系数小于 0.05 的项，可降维至 109.)
% Iterate to remove terms with coefficients < 0.05, B dimension -> 109;
% Iterate to remove terms with coefficients <    0.1, B dimension ->    31;
% Iterate to remove terms with coefficients < 0.15, B dimension ->    13;
% Iterate to remove terms with coefficients <    0.2, B dimension ->     4.
last_B_dim = 0;

fprintf('Start iteration to reduce dimension, min_coef = %g:\n',min_coef)
while 1
    for j = length(B) : -1 : 2 % Ignore column 1(constant term).
        if abs(B(j)) < min_coef
            % Remove terms with coef less than min_coef.
            X(:,j) = [];
            % Remove corresponding columns in X_test.
            X_test(:,j) = [];
        end
    end
    fprintf('Now X has %d columns.\n',size(X,2)) % DEBUG
%--------------------
%       [B,~,~,~,~] = regress(Y,X);
    X(:,1) = [];
    [B,STATS] = robustfit(X,Y,'logistic');
    X = [ones(num_samples - 200,1),X];
%--------------------
    Yhat = X * B;

    % break condition: the dimension of stops decreasing.
    if last_B_dim == length(B)
        break;
    end
    last_B_dim = length(B);
end
disp('Iteration finished.')
clear last_B_dim
%----------------------------------------
end
%------------------------------------------------------------
end
```

```
%%
% Calculate accuracy of discrimination using 200 test samples.
p2_s3_calc_correct_pct;

%%
% Save to .mat file.
if p2_save_result
%-------------------------------------------------------------
% str1:extract_method; str2: threshold
switch reg_method
	case 1
		str_method = '_regress_';
	case 2
		str_method    = '_stepwise_';
	case 3
		str_method = '_logistic_';
end

if iterate
	str_iterate = '_iterate';
else
	str_iterate = [];
end

filename = ['p2_result_mat/p2_result_' str1 str_method str2 str_iterate '.mat'];
save(filename,'threshold','B','healthy_test','healthy_test_correct_pct',...
	'ill_test','ill_test_correct_pct','healthy_training','healthy_training_correct_pct',...
	'ill_training','ill_training_correct_pct','STATS')

fprintf('Save to file %s.\n',filename)
disp('==========================================')
%-------------------------------------------------------------
end
clear p2_save_result

%%
clear str1 str2 filename str_method reg_method iterate
```