厦門大學

本 科 毕 业 论 文 （设 计）

（主修专业）

基于单细胞基因测序数据的分类问题研究

Research on Classification Problem Based on Single Cell Gene Sequencing Data

姓　　名：王力为

学　　号：19020152203156

学　　院：数学科学学院

专　　业：统计学

年　　级：2015 级

校内指导教师：胡杰　助理教授

二〇　一九　年 四 月 二十八 日

厦门大学本科学位论文诚信承诺书

本人呈交的学位论文是在导师指导下独立完成的研究成果。本人在论文写作中参考其他个人或集体已经发表的研究成果，均在文中以适当方式明确标明，并符合相关法律规范及《厦门大学本科毕业论文（设计）规范》。

该学位论文为（　　）课题（组）的研究成果，获得（　　）课题（组）经费或实验室的资助，在（　　）实验室完成（请在以上括号内填写课题或课题组负责人或实验室名称，未有此项声明内容的，可以不作特别声明）。

本人承诺辅修专业毕业论文（设计）（如有）的内容与主修专业不存在相同与相近情况。

学生声明（签名）：

年　　月　　日

致谢

时间飞逝，大学四年的生活转眼已经接近尾声，在数院的四年大学生活让我受益匪浅。历经多个月的努力，我的毕业论文终于在各位老师同学的帮助下完成了。回顾整个毕业论文的完成过程，从确定论文题目、阅读文献、搜集资料、编写代码、论文写作到最终完成，我都得到了很多的帮助和关心，我想要在这里表达对他们的感谢。

首先，我想要感谢我的导师胡杰老师。本论文的选题以及写作指导过程要十分感谢从大三上学期开始，带我开始着手做该课题，在我的课题上帮助我解决了很多的问题，包括如何从课堂上和看课外论文的问题、转化成一个讨论班的project、最后扩宽思路，慢慢成型为我人生第一篇学术论文，成就感满满，也在此谨向胡杰老师致以诚挚的谢意和崇高的敬意。

其次，我要感谢大学四年以来，所有本专业的老师在数理逻辑和统计知识上对自己的培养，也包括双学位两年来所有任课老师，在计量和统计的应用上对自己的教导以及耐心仔细的辅导帮助，特别是教我数据挖掘课程的方匡南教授，使得我在大学的知识面上扩充了自己的的广度以及深度，得到了很多的收获。

最后，我要感谢整个数院和厦门大学，让我在这里认识到了很多的良师益友，充实而快乐的度过了大学四年的美好时光。

在今后的学习生活中，我将铭记各位老师同学对我的帮助，我将会继续努力，用更好的成绩来回报帮助过我的人。

摘要

本文从分类问题和变量筛选的角度，尝试识别了造成两组单细胞病变与否的关键突变基因。基于非平衡+高维+零膨胀+小样本的单细胞RNA基因测序的表达数据，本文采取了如下几个步骤进行分析。首先，本文采取正态分布检验的方法预筛剔除了过度零膨胀及不具有生物学意义的基因变量。进一步，本文使用shrinkage+Logistics二分类方法，即Lasso以及其变型来筛选关键基因变量。同时，本文选择使用合成取样SMOTE方法生成新的训练数据以改善数据的非平衡性，有效提升了分类器的AUC值及变量筛选结果。找到显著影响细胞病变的潜在关键基因变量后，本文将选出的备选基因变量分别做了T分布均值假设检验，及多重假设检验来进行二次验证，得到最终关键基因结果。最后，本文参考了网上生物学和基因的背景知识，对最终结果进行了一定的解释。

关键词 非平衡数据；SMOTE；高维数据变量筛选；Lasso及其变型；多重假设检验

Abstract

From the perspective of classification problems and variable selection, this thesis attempts to identify key mutation genes that cause two groups of single-cell lesions. Based on the expression data of unbalanced, high-dimensional, zero-expansion with small sample single-cell RNA gene sequencing data, the following steps were taken for analysis. First, this thesis adopts the method of normal distribution test to pre-screen out genetic variables with excessive zero expansion and no biological significance. Further, this thesis uses the shrinkage and Logistics binary classification method, i.e, Lasso and its variants, to select key genetic variables. At the same time, this thesis chooses to use the synthetic sampling method, SMOTE, to generate new training data to improve the balance of previous unbalanced data, and effectively improve the AUC value of the classifiers and the variable selection results. After finding potential key genetic variables that significantly affect cytopathic changes, the candidate gene variables selected in this thesis were subjected to T-distribution mean hypothesis test and multiple hypothesis tests for secondary verification to obtain the final key gene results. Finally, this article refers to the background knowledge of biology and genes online, and explains the final results.

Keywords Unbalanced Data; SMOTE; Variable Selection on High-Dimensional Data; Lasso; Multiple Hypothesis Testing

目录

Contents

第1章　引言

1.1　基因突变与单细胞RNA基因测序

基因是遗传的基本单位，每一个基因都能够通过编码，储存一段用于合成特定蛋白质的讯息。染色体则是基因的主要载体，人类细胞至少有3 万个基因的DNA 编码储存其中。在生物的生命活动中，所有的基因都履行各自的职能，通过“转录”与“翻译”行为，合成其对应的蛋白质。一部分蛋白质可能会作为生命必要组织的组成成分。一部分蛋白质则扮演着调节细胞生长、分裂、分化、凋亡的角色。这种蛋白质，由基因遗传信息精确掌控，不能太多也不能太少。如果这些基因出了差错，细胞的生命流程可能就会出现问题。

所谓基因出现问题，实际上也就是基因发生了突变。而准确识别这些突变基因或关键基因也正是生物医学上不断攻克的一大难点。例如，对于罹患同种癌症的不同患者，其体内肿瘤细胞的外观、遗传信息都可能有所不同。而其最核心的原因就是基因的表达情况有所差异。如果我们能够准确识别这些突变基因，以及了解它们的差异性，则可以在分子水平上，基于不同患者的基因表达进行调控，来完成精准医疗。

单细胞RNA基因表达分析（scRNA-seq）是近年来兴起的技术，正在彻底改变整个生物体科学，也为解决关键突变基因识别问题提供了帮助。该技术可以在细胞水平上无偏地识别先前未鉴定过的分子异质性，并提供了解剖复杂组织和特定细胞环境组成的手段。由于与传统的bulk测序技术相比，单细胞RNA测序数据可以通过单细胞水平上基因表达的分子表征来洞察正常的细胞功能和各种疾病状态，可以看到单细胞水平的多样性——如同一类细胞的不同发育情况等，因此也可以更好确定某些显著的关键基因。

1.2　课题的研究背景

在本文中，课题的研究背景正是单细胞RNA基因测序与基因突变。实验主角是中性粒细胞（单细胞）。中性粒细胞生长发育有很多阶段，如成熟或不成熟（成熟没有分化能力、有杀菌能力）。同样是不成熟的中性粒细胞，也有很多区别。在本次实验里，中性粒细胞经过某种感染后，全部“病变退化”成不成熟细胞，而且还和传统不成熟的“正常”细胞不一样。我们感兴趣的就是这一部分“病变”的未成熟细胞，和未感染前“正常”的未成熟细胞的区别。造成两组细胞区别的原因，就在于某些关键基因突变，表达数据产生显著变动。因此，我们需要找出这部分关键基因，了解正常和病变细胞的不同之处。

本文数据来自于一个小鼠胃溃疡实验，对照组（sham）是未受感染的小鼠机体脾脏的正常未成熟中性粒细胞（以后简称正常细胞）；实验组（CLP）是受到某种感染产生病变的未成熟中性粒细胞（以后简称病变细胞）。看对照小鼠和实验小鼠中性粒细胞RNA的测序基因表达，在有无刺激下面，会不会有某些基因是显著的不同。

利用生物统计及机器学习的知识，我们可以从分类问题和变量筛选的角度去尝试解决识别关键突变基因的问题。然而，单细胞RNA基因测序数据的特点一般是：超高维、非平衡、零膨胀严重/不满足正态分布/分布不均衡、小样本，还有潜在的噪声、不符合IID样本数据的假设等。以本文数据集为例，对照组样本个数大大少于实验组个数，数据非平衡问题严重；而总样本量也仅有420个，与上万基因自变量比起来的比例约为1:40。一般来说，把问题根据难度从小到大排序有：大数据+分布均衡<大数据+分布不均衡<小数据+数据平衡<小数据+数据非平衡<小数据+数据非平衡+超高维问题。对此类数据进行正确有效的分类，并且找到影响分类的关键基因，整体难度不小。

基于现有条件及难点问题，本文在此对其进行逐一分析：

（1）首先，根据实验组与对照组的区分，以及标签（Condition）已知，我们假定这是一个有监督的二分类问题。

（2）对于分类问题，小样本的特性决定了不能使用太过于复杂的分类算法，比如神经网络等。

（3）零膨胀/不满足正态分布的问题可能会影响分类问题的估计，要进行一定的处理。本文处理方法为不满足条件的变量的预筛删除。

（4）非平衡和超高维数据的问题则为本文探讨的重点，将逐一讨论。

（5）数据噪声及IID假设则将在后续继续深入的思路部分的文献方法中提到。

以下是本文主要工作的概述：

- 探讨和尝试各类方法，解决了数据为非平衡+高维+零膨胀+小样本的二分类变量筛选问题，找到显著影响分类的关键基因变量。

- 本文主要使用shrinkage+Logistics类方法，即Lasso以及其变型来筛选变量。最后将选出的备选基因变量做多重假设检验来进行二次验证，得到最终关键基因结果。同时，本文使用正态性检验方法预剔除零膨胀严重的基因变量，并使用合成取样SMOTE方法生成新的训练数据以改善数据的非平衡性，有效提升了分类器的AUC值及变量筛选结果。

- 从数据中得到了关键基因的结果后，参考了生物学和基因背景知识进行解释。

第 2 章　方法综述

本文分析的主要解决的问题是非平衡与高维变量筛选及降维问题。为了方便后面章节的分析和应用，该部分将会对本文运用到的方法依次进行综述。

2.1　非平衡问题

非平衡数据的分类学习是机器学习和数据挖掘中的重要问题。在非平衡数据的状况下，分类器能够在多数类上面有很好的准确率，但是在少数类上准确率却很糟糕，主要是因为更大的多数类在传统训练标准上面的影响。在非平衡问题不解决前，大部分分类方法，如所有树方法、集成方法、SVM分类方法表现都是很糟糕的。如果数据存在严重的不平衡，预测得出的结论往往也是有偏的，即分类结果会偏向于较多观测的类。对于这种问题该如何处理呢?

2.1.1　常见方法

1 采样方法

采样方法是通过对训练集进行处理使其从不平衡的数据集变成平衡的数据集，在大部分情况下会对最终的结果带来提升。

随机采样分为过采样（Oversampling）和欠采样（Undersampling）。过采样是把小种类复制多份，如进行Bootstrap 抽样。欠采样是从大众类中剔除一些样本，或者说只从大众类中选取部分样本。

随机采样最大的优点是简单，但缺点也很明显。过采样后的数据集中会反复出现一些样本，训练出来的模型会有一定的过拟合；而欠采样的缺点显而易见，那就是最终的训练集丢失了数据，模型只学到了总体模式的一部分。

2 一分类

对于正负样本极不平衡的场景，我们可以换一个完全不同的角度来看待问题：把它看做一分类（One Class Learning）或异常检测（Novelty Detection）问题。这类方法的重点不在于捕捉类间的差别，而是为其中一类进行建模，经典的工作包括One-class SVM等。

然而，One-class SVM不具备变量选择的功能。因此本文将采用采样的方法对非平衡数据进行处理。

2.1.2 合成取样SMOTE方法

为了解决数据的非平衡问题，Chawla等(2011)提出了SMOTE算法，即合成少数过采样技术，它是基于随机过采样算法的一种改进方案。该技术是目前处理非平衡数据的常用手段，并受到学术界和工业界的一致认同。

SMOTE（Synthetic Minority Oversampling Technique），合成少数类过采样技术．它是基于随机过采样算法的一种改进方案，由于随机过采样采取简单复制样本的策略来增加少数类样本，这样容易产生模型过拟合的问题，即使得模型学习到的信息过于特别(Specific)而不够泛化(General)，SMOTE算法的基本思想是对少数类样本进行分析和模拟，并根据少数类样本人工合成新样本添加到数据集中，进而使原始数据中的类别不再严重失衡。

具体如下图所示，算法流程如下。

1 对于少数类中每一个样本x，以欧氏距离为标准计算它到少数类样本集S_{min}中所有样本的距离，得到其k近邻。

2 根据样本不平衡比例设置一个采样比例以确定采样倍率N，对于每一个少数类样本x，从其k近邻中随机选择若干个样本，假设选择的近邻为$\hat{x}_i$。

3 对于每一个随机选出的近邻$\hat{x}_i$，分别与原样本按照如下的公式构建新的样本

$$x_{new} = x_i + (\hat{x}_i - x_i) \times \delta$$

其中$\delta \in [0, 1]$是随机数。

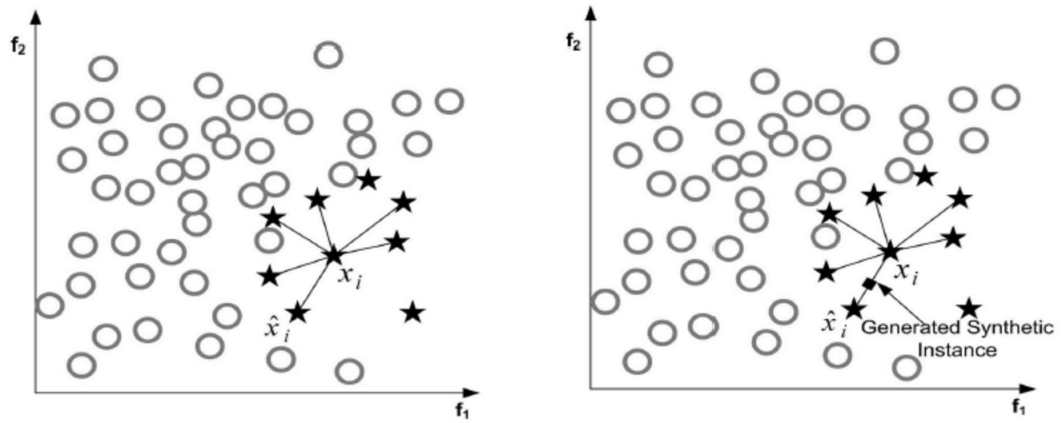

图 2-1 SMOTE原理图例

上图是SMOTE方法在$K = 6$近邻下的示意图，黑色星格是生成的新样本。

2.2　高维数据变量筛选问题

在机器学习和统计学中，特征选择也被称为变量选择、属性选择或变量子集选择。它是指：为了构建模型而选择相关特征（即属性、指标）子集的过程。特征选择的目标是寻找最优特征子集。特征选择能剔除不相关(irrelevant)或冗余(redundant)的特征，从而达到减少特征个数，提高模型精确度，减少运行时间的目的。另一方面，选取出真正相关的特征能够简化模型，能协助理解数据产生的过程。

在基因相关性研究中，相比起几百个细胞的样本量，基因的个数往往远大于样本的个数，基因数据维度动辄上万，而实际对特定表型特征有所影响的通常只有少数。因此，为了避免维数灾难与过拟合，需要对参数进行适当的特征选择。

按照评价标准的不同，常见的特征选择方法可以大致分为三类：过滤类法(filter)、包装类(wrapper)、嵌入类(embedding)。

包装类需要对所有特征子集进行模拟（如逐步回归），在高维参数空间难以实现。

而过滤类方法（如利用相关系数的方法进行预筛），主要侧重于单个特征跟目标变量的相关性，优点是计算时间上较高效,对于过拟合问题也具有较高的鲁棒性，缺点就是倾向于选择冗余的特征,因为他们不考虑特征之间的相关性,有可能某一个特征的分类能力很差，但是它和某些其它特征组合起来会得到不错的效果，因此，过滤类的方法往往准确度低于包装类。

嵌入式特征选择方法是将特征选择过程与学习器训练过程融为一体，两者在同一个优化过程中完成，即在学习器训练过程中自动完成了特征选择（如Lasso和决策树）。因此本文采用嵌入类方法。

从常见方法的角度来说，对高维数据进行变量筛选的方法有如下几类：

1 逐步回归法。它的优点是考虑了特征与特征之间的关联性，缺点是：当观测数据较少时容易过拟合，而当特征数量较多时,计算时间又会增长。本文不用逐步回归法是因为基因自变量的维度太大，逐步回归法极易陷入局部最优解。

2 主成分法。但本文也不打算使用，原因是因为变量筛选基因时，要具体到某一个特定的基因，而不是某一组基因，所以主成分出来的意义不大。当然除非该基因组有非常强的生物学解释意义，可尝试这种做法。

3 决策树（带剪枝）。决策树用于变量选择时，在生物基因表达数据上拥有更加便于解释的优点。但在非平衡问题上缺点暴露地一览无遗，且易受噪声数据影响结果。

4 基于树模型的集成方法：如RandomForest、Adaboost，可按照Gini系数计算的变量重要性排序选取的基因加入参考。然而该类方法在非平衡数据上表现仍然不佳。

5 各种shrinkage压缩方法+ LR(Logistics)方法。本文主要采用该类方法，主要使用了Lasso及其变型，如elastic net，adaptive Lasso等方法，以及SCAD及MCP法。

本文主要使用是shrinkage压缩类方法，即Lasso以及其变型。最后将选出的基因变量进行多重假设检验验证筛选，并结合生物学知识进行解释。

2.2.1 LASSO及其变型

（1）嵌入类方法最为知名的无疑是LASSO 方法(Tibshirani，1996)，通过引入惩罚项L_1 范数 $\lambda|x|$对参数进行惩罚，其中$\lambda \geq 0$ 是控制惩罚的力度，亦或者是特征矩阵的稀疏程度。LASSO 通过L_1 惩罚的方式，使得那些不太重要的特征系数变为0，从而达到特征选择的效果。

$$\arg\min_{\beta} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| = \mathrm{RSS} + \lambda \sum_{j=1}^{p} |\beta_j|$$

或

$$\arg\min_{\beta} \left\{ \|Y - X\beta\|_2^2 \right\} \text{ subject to } \lambda \|\beta\|_1 \leq t$$

（2）但当同时有多个特征彼此高度相关时，这些方法更倾向于只选择其中一个参数。对此问题，将岭回归L_2 范数与LASSO 进行结合的的Elastic Net Regularization 方法，能够倾向于从高度相关的多种特种中选择两种(Zou and Hastie，2005)，

Elastic Net 的目标函数：

$$\arg\min_{\beta \in \mathbb{R}^p} \left\{ \|Y - X\beta\|_2^2 + \lambda_1 \|\beta\|_1 + \lambda_2 \|\beta\|_2^2 \right\}$$

或

$$\arg\min_{\beta_0, \beta} \left\{ \|Y - \beta_0 - X\beta\|_2^2 \right\} \text{ subject to } (1-\alpha)\|\beta\|_1 + \alpha \|\beta\|_2^2 \leq t$$

$$\text{where } \alpha = \frac{\lambda_2}{\lambda_1 + \lambda_2}, 0 \leq \alpha \leq 1$$

（3）而另一方面，上述提到的各种LASSO 或基于LASSO 的算法,都有过度压缩非零系数的情况, 这使得得到的参数估计不具有相合性。对此，Zou（2006）提出了自适应的Adaptive Lasso 方法，将LASSO 中的惩罚项进行了修正，使其在变量个数维持不变时，其估计结果具有渐进相合性。

对所有加权Lasso：

$$\arg\min_{\beta}\left\{\|Y-X\beta\|^2+\lambda\sum_{j=1}^{p}w_j\,|\beta_j|\right\}$$

其中，当权重取为

$$\hat{\mathbf{w}}=1/|\hat{\boldsymbol{\beta}}|^{\gamma}$$

有

$$\hat{\boldsymbol{\beta}}^{*(n)}=\arg\min_{\beta}\left\{\|Y-X\beta\|^2+\lambda_n\sum_{j=1}^{p}\hat{w}_j\,|\beta_j|\right\}$$

一般γ会取为1。γ越大，渐进相合速度越快。

可证明Adaptive Lasso方法筛选出来的参数$\hat{\boldsymbol{\beta}}^{*(n)}$满足渐进相合性。

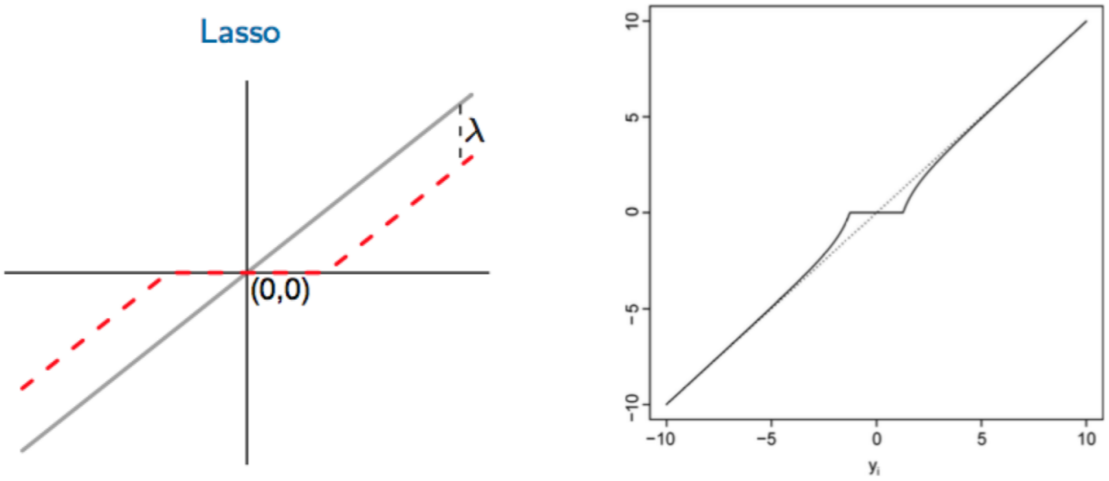

图 2-2 Lasso与Adaptive Lasso方法的渐进相合性对比。$\gamma=2$。其中灰色实线表示在标准化数据下，无偏$\hat{\boldsymbol{\beta}}_{OLS}$的相对位置。Adaptive Lasso在绝对值较大的系数上满足相合性

2.2.2　SCAD与MCP

（1）Fan and Li (2001) 提出了有别于L_1、L_2范数的SCAD方法及其惩罚项。这种方法的惩罚函数是对称且非凹的，并且可处理奇异阵以产生稀疏解。这种方法具有广泛的适用性，可以应用于广义线性模型，强健的回归模型。

Smoothly Clipped Absolute Deviation (SCAD) penalty：

$$P(\beta|;\lambda,a)=\begin{cases}\lambda|\beta| & \text{if } 0\le|\beta|<\lambda\\ -\frac{\beta^2-2a\lambda|\beta|+\lambda^2}{2(a-1)} & \text{if } \lambda\le|\beta|<a\lambda\\ (a+1)\lambda^2/2 & \text{otherwise}\end{cases}$$

更进一步地，SCAD penalty也具有Oracle性质，即具有渐进相合性。

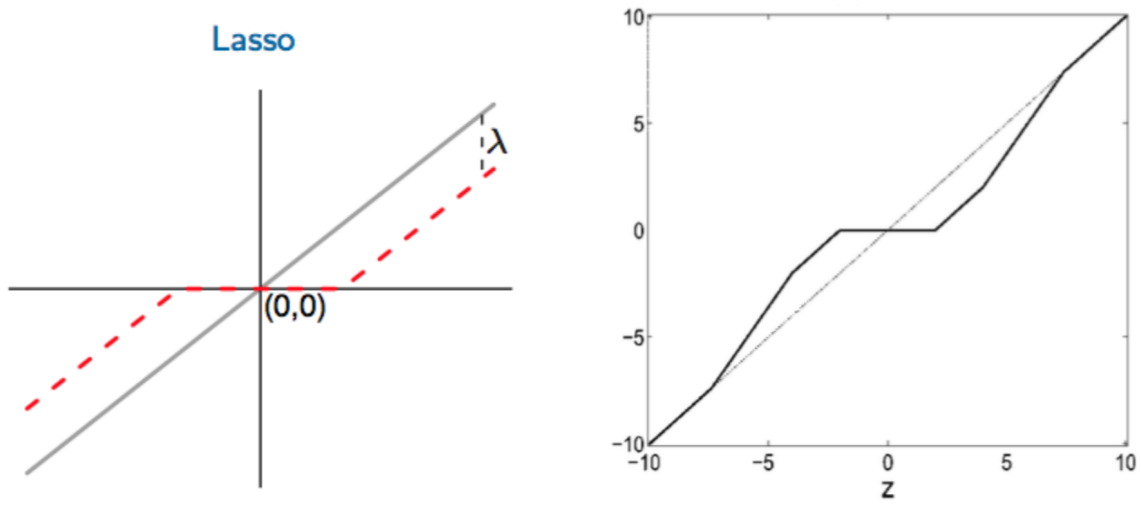

图 2-3 Lasso与SCAD方法的渐进相合性对比，SCAD参数$\lambda = 2, a = 3.7$

从图2-2和2-3可以看到，lasso压缩系数是始终有偏的，而Adaptive Lasso和SCAD既能连续的压缩系数，也能在较大的系数取得渐近无偏的估计。

（2）Zhang等(2010)基于SCAD penalty提出了新的惩罚项MCP，也具有渐进相合性的优良性质。

Minimax Concave Penalty (MCP)：

$$P(\beta|;\lambda,a) = \begin{cases} \lambda|\beta| - \frac{|\beta|^2}{2a}, \text{ if } |\beta| \le a\lambda \\ \frac{a\lambda^2}{2}, \text{ if } |\beta| > a\lambda \end{cases}$$

然而，SCAD和MCP的缺点也较为明显。由于模型形式过于复杂，迭代算法运行速度较慢，因此该方法在噪声数据量较小的情况下表现较优，但在噪声数据量较大的的情况下表现较差。

所有方法皆适用于用于广义线性模型，包括Logistics回归。本文将采用MCP作为其中一种方法进行变量选择，与Lasso族方法进行对比，应用在Logistics回归中，比较不同惩罚项对高维二分类问题的效果差异。

2.3 降维问题

当处理真实问题和真实数据时，我们往往遇到维度高达数百万的高维数据。尽管在其原来的高维结构中，数据能够得到最好的表达，但有时候我们可能需要给数据降维。简而言之，降维就是用2维或3维表示多维数据（彼此具有相关性的多个特征数据）的技术，利用降维算法，可以显式地表现数据。

2.3.1　常用降维算法概述

常见方法有以下几种。

1 主成份分析（Principal Component Analysis，PCA）（线性）

2 局部线性嵌入(Locally-linear embedding，LLE)（非线性）

3 拉普拉斯特征映射（Laplacian eigenmaps，LE）（非线性）

4 等距映射Isomap（非线性）

5 随机邻域嵌入Stochastic Neighbor Embedding（SNE）及其改型t-SNE（非线性）

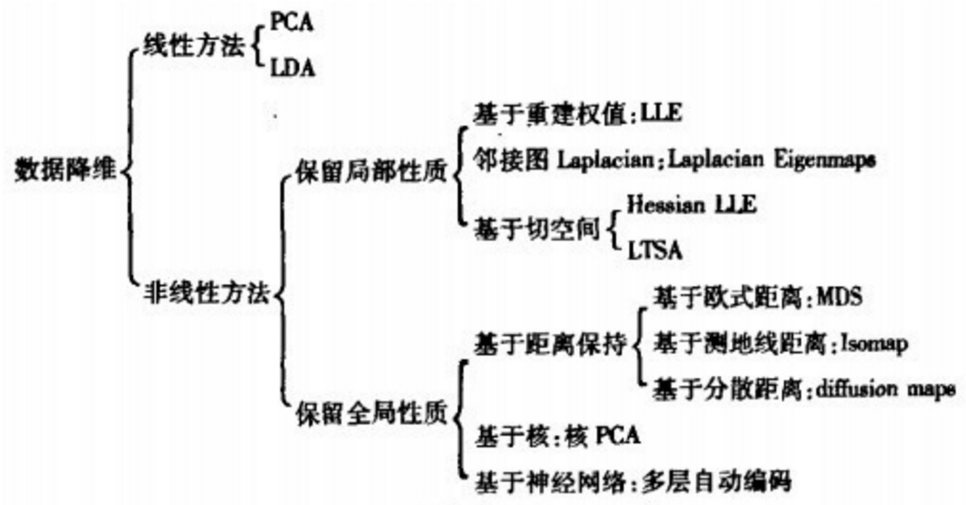

图 2-4 降维问题常见方法分类

本文将注意力放在了探讨我们所学过的线性算法PCA和本文采用的非线性算法tSNE的区别。

PCA是一种线性算法，传统而有效，关于PCA的算法原理本文不进行赘述。然而线性算法的投影原理（projection）不总是降维最好的方法。比如在许多情况下，空间可以扭转，因此它不能解释特征之间的复杂多项式关系。而基于流行数据进行建模的降维算法称为流形学习（Manifold Learning）。它假设大多数现实世界的高维数据集接近于一个低维流形，空间可以扭转，因此流行学习保留下的属性信息，更具代表性，也即最能体现样本间的差异。

2.3.2　TSNE非线性降维

Maaten and Hinton（2008）提出了TSNE降维方法。TSNE（t-分布随机邻域嵌入，t-Distribution Stochastic Neighbor Embedding），是一种用于探索高维数据的非监

督非线性的降维算法，它将多维数据映射到适合于人类观察的两个或多个维度。t-SNE是由SNE衍生出的一种算法。SNE最早出现在2002年，它改变了流行学习中MDS和ISOMAP中基于距离不变的思想，将高维映射到低维的同时，尽量保证相互之间的分布概率不变。因此在介绍TSNE前，我们先来介绍一下SNE:

SNE是通过仿射(affinitie)变换将数据点映射到概率分布上，主要包括两个步骤:

（1）SNE构建一个高维对象之间的概率分布，一般是正态分布，使得相似的对象有更高的概率被选择，而不相似的对象有较低的概率被选择。

（2）SNE在低维空间里在构建这些点的概率分布，同样也是正态分布，使得这两个概率分布之间尽可能的相似。具体见下:

随机邻接嵌入（SNE）通过将数据点之间的高维欧几里得距离转换为表示相似性的条件概率而开始。数据点x_i、x_j之间的条件概率$p_{j|i}$由下式给出，表示第i个样本分布在样本j周围的概率，使用正态分布计算:（注:$p_{i|i}=0$）

$$p_{j|i}=\frac{\exp\left(-\left\|x_i-x_j\right\|^2/2\sigma_i^2\right)}{\sum_{k\neq i}\exp\left(-\left\|x_i-x_k\right\|^2/2\sigma_i^2\right)}$$

其次是构建低维空间的概率分布，低维数据用y表示:

$$q_{j|i}=\frac{\exp\left(-\left\|y_i-y_j\right\|^2\right)}{\sum_{k\neq i}\exp\left(-\left\|y_i-y_k\right\|^2\right)}$$

使用KL散度构建损失函数:

$$C=\sum_i KL\left(P_i\|Q_i\right)=\sum_i\sum_j p_{j|i}\log\frac{p_{j|i}}{q_{j|i}}$$

计算损失函数梯度:

$$\frac{\partial C}{\partial y_i}=2\sum_j\left(p_{j|i}-q_{j|i}+p_{i|j}-q_{i|j}\right)\left(y_i-y_j\right)$$

随机给定一个初始化的Y，进行优化，采用梯度下降方法最小化损失函数，使得Y的分布矩阵逼近X的分布矩阵，以此确定低维空间的概率分布的点，使得高维空间概率分布和低维空间概率分布的差异最小，以此尽可能相似:

$$y^{(t)}=y^{(t-1)}+\eta\frac{\partial C}{\partial y_i}+\alpha(t)\left(y^{(t-1)}-y^{(t-2)}\right)$$

这个算法有两个输入，一个是数据本身，另一个被称为困惑度（perplexity）。更高的困惑度意味着一个数据点会把更多的数据点看作是其紧密的近邻点，更低的困惑度就更少，它与分布的σ_i的计算有关，其中σ_i是以数据点x_i为中心的正态方差。

在此基础上，TSNE引入对称SNE思想，让高维和低维中的概率分布矩阵是对称的。该方法先使用对称条件概率来定义一个联合概率

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$$

使得$p_{ij} = p_{ji}$，且$p_{ii} = 0$。同时我们希望低维概率也满足$q_{ij} = q_{ji}$，能方便运算。

同时，低维中的分布则采用更一般的T分布，这样做的好处是为了让距离大的簇之间距离拉大，从而极大改良了可视化后数据簇的拥挤问题：

$$q_{ij} = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \|y_k - y_l\|^2\right)^{-1}}$$

损失函数的构建方法不变。则计算损失函数梯度变为：

$$\frac{\partial C}{\partial y_i} = 4 \sum_j \left(p_{ij} - q_{ij}\right)\left(y_i - y_j\right)\left(1 + \|y_i - y_j\|^2\right)^{-1}$$

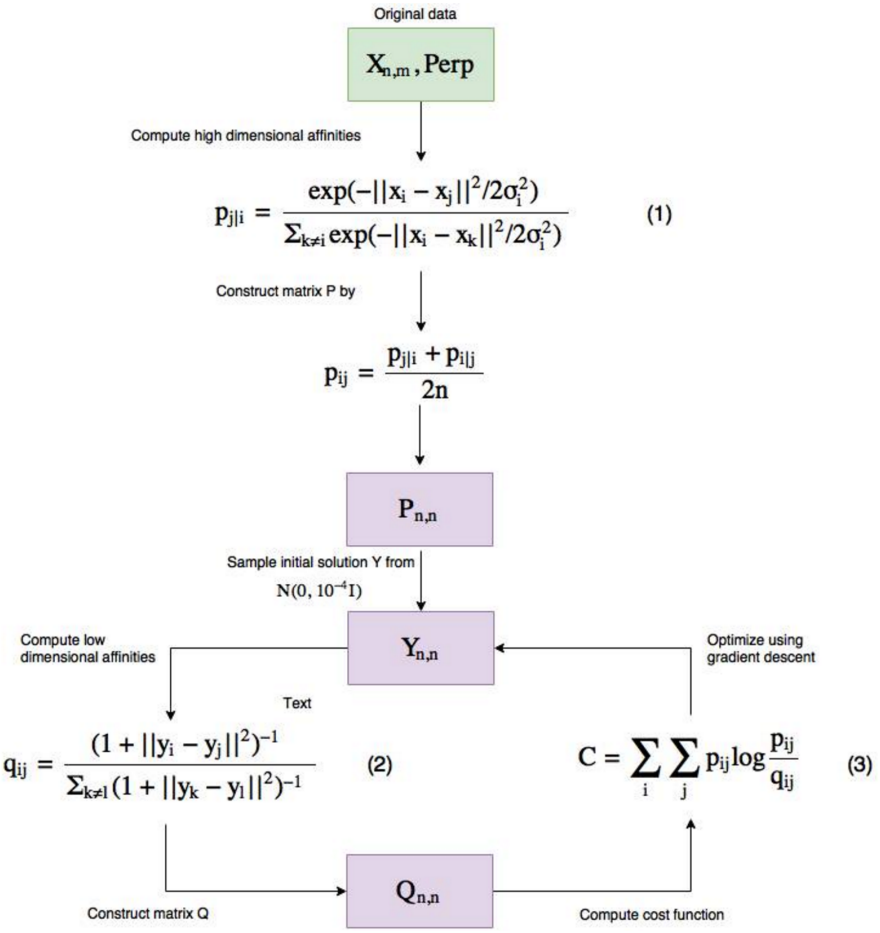

图 2-5 t-SNE过程流程总结图

2.4 多重假设检验FDR

在很多科学实验中，在某些情况下，我们要做多次判断。然而，一个小效率事件的假设检验，就在多次反复尝试后，变成了一个多次出现的事，即在进行多次检验后（也就是所说的多重检验，multiple test），那么基于单次比较的检验标准将变得过于宽松，使得阳性结果中的错误率（FDR值）已经大到令人不可忍受的地步。

如何做出修正? 最好的办法就重新提高判断的标准（p value），使得单次判断的犯错概率就会下降，那么总体犯错的概率也将下降。在多重检验中提高判断标准的方法，我们就称之为“多重检验校正”。对统计分析结果产生的P值做一次筛选，对原始P值这个显著性的真实性用FDR校正后的新P值做判定。

在多重假设检验方法中，最简单严厉的方法要属于Bonferroni校正。Bonferroni法如何校正呢? 用一个例子来说明：假设原来使用p value为1 %的标准判断是否差异表达，结果对于10000个没有差异的基因也会错误地得出“100个基因差异表达的结论”。那么，若将p value阈值直接提高到1×10-6，同样的10000次比较之后，平均假阳性次数也依然被控制在0.01次，足够严格了。但这依然有一个问题，标准定太高了，如果一些基因真的存在表达差异，也很有可能达不到我们的阈值标准，被误判为没有差异，这就是假阴性率提高了。

目前在RNA-seq中，使用最普遍的是Benjamini and Hochberg（1995）第一次提出的FDR(False Discovery Rate)的概念以及相应的多重检验校正方法。FDR（假阳性率）错误控制法基本原理是通过控制FDR值来决定P值的值域。相对Bonferroni来说，FDR用比较温和的方法对p值进行了校正，其试图在假阳性和假阴性间达到平衡。

在前面的例子的10000次基因差异比较中，如果我们使用FDR为1%的标准进行检验，最后检测出显著差异（阳性结果）的基因数是100个，那么其中假阳性的个数就可以被控制在1个，剩下的99个则是真实的差异（FDR被控制在1%）。

使用FDR的控制方法计算出的被校正后的p value称为Q value,计算方法如下：

$$q_i = p_i \times \frac{n}{rank_i}$$

其中n为假设检验的p-value个数。$rank_i$是指p-value从小到大排序后的次序，最大的p-value的rank值为n，第二大则是n-1，依次至最小为1。

在国内目前大部分公司提供的RNA-seq类的生物信息分析中（包括基迪奥公司提供的分析服务），都会使用多重检验校正，而且基本都会使用Benjamini在1995年报道的FDR校正方法。

第 3 章　数据说明与预处理

3.1　原始数据说明

本次分析本次实验单细胞测序的基因表达数据，420个细胞是单细胞，共16926个基因。402个非正常实验组（病变），即CLP，18个对照组（正常），即sham。合起来是一个420*16927的高维宽矩阵。数据无缺失值。具体变量说明如表1 所示。

表 1：数据说明表

$n = 420$	变量名	详细说明	取值范围	变量数	其他
因变量	Condition	二分类数据	正常与病变	1	正常：病
自变量	NDC80等	连续型数据	≥0的实数	16926	变=18:402

对420个细胞的全部16926个基因表达数值的均值和标准差计算如图1所示。其中三角形点为少数类样本（正常细胞，sham）。由上表和下图都可以看出数据的非平衡性。

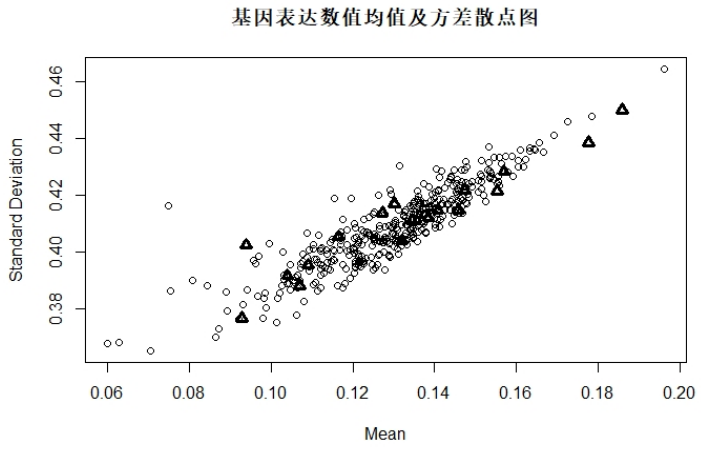

图 3-6 原始数据420个样本的基因表达数值均值及方差散点图

3.2　数据预处理与划分

3.2.1　正态性检验筛选基因自变量

即便本文将应用的Lasso方法可以起到变量筛选的作用，但在待估计参数维度远高于样本总数时，Lasso不能展现太好的效果。本文的数据矩阵就是个小数据大样本，420*16927的高维宽矩阵，因此进行基因自变量的预筛，该预处理过程也非常重要。

于是，本文通过正态性检验的方法和生物学的先验知识，对基因自变量进行了初步筛选，再将初步筛选后的基因放入模型中计算。

首先我们进行Shapiro-Wilk正态性检验。检验的原假设是该自变量数据分布满足正态分布。依次对16926个基因自变量做检验，将P值从大到小排列，理由是P值越大，该自变量数据分布满足正态分布的结论越可靠；反之，该基因自变量越不满足正态分布。

我们发现，基因数据分布在P值大小排在第635位及其之后的对应基因，与之前的基因数据分布上出现了较大差别（即右图）：

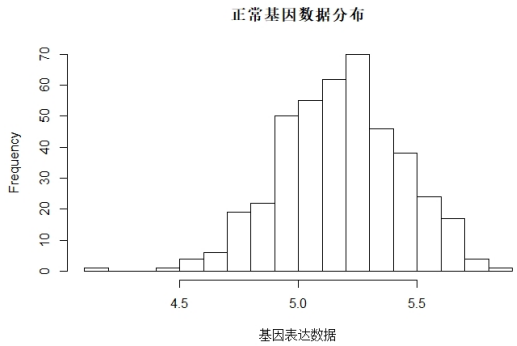

图 3-7 正常基因

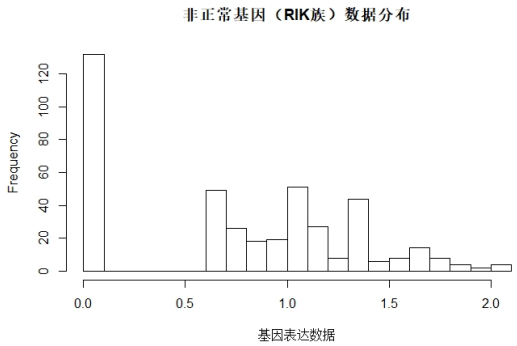

图 3-8 非正常基因(RIK族)

同时，从原始数据中可以发现，从第635位开始的基因数据皆来源于RIK族基因。我们有一定理由认为该类基因数据零膨胀严重，严重不满足正态分布，有筛除的必要。

不过，由于Shapiro-Wilk正态性检验对原始分布是正态分布的假设要求很严格，且对过小样本（$n < 8$）和过大样本（$n > 50$）不太有效，而这里的样本量已经达到了$n = 420$，所以我们再进行Kolmogorov-Smirnov非参数正态性检验来验证我们的结果。检验的原假设依然是该数据分布满足正态分布。

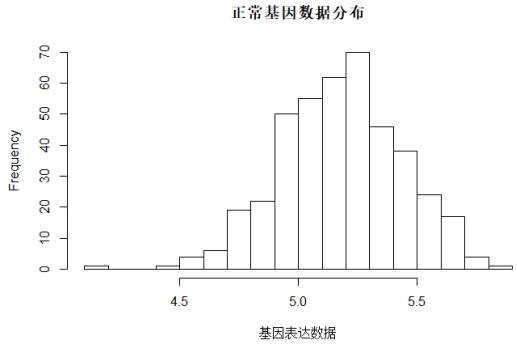

图 3-9 正常基因

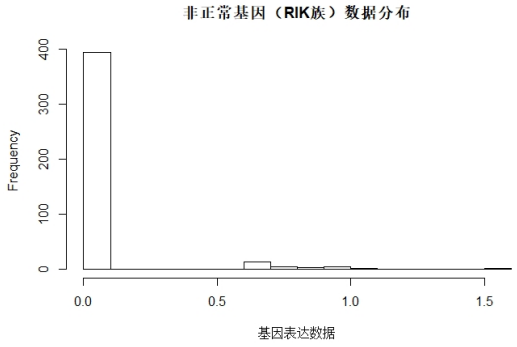

图 3-10 非正常基因(RIK族)

同样，由左图分布图可知，前634个基因近似正态，而从第635个基因开始，全部为RIK族基因，从数据分布（即右图）来看，它有过度零膨胀、不满足正态分布的特

点，且K-S检验的P值排序对应结果更加明显。同时，根据生物学先验知识，可知RIK基因不具有生物学意义。因此该类RIK基因作为不正常基因，从自变量数据中全部筛除。

自变量剩下634个维度，因此X成为420*634的矩阵，数据合为420*635的矩阵。

3.2.2　训练集与测试集的划分处理

将原始数据进行基因自变量的预筛后，可开始对数据进行划分，拆分成训练集和测试集。本文采用分层拆分，分别把多数类样本和少数类样本两部分对半分，二分之一分给测试集，二分之一为训练集。此时以测试集为例，共有210个样本，其中对照组sham（少样本）仅有9个。划分说明如表2：

将少数类对照组样本sham（正常）设为0，多数类实验组样本CLP（非正常）设为1。

表 2 ： 训练集与测试集数据划分说明表

	样本数n	少数类（0）个数	多数类（1）个数	自变量个数
训练集	210	9	201	634
测试集	210	9	201	634

3.2.3　生成SMOTE仿真数据

进行基因的预筛缓解了超高维问题后，我们继续使用SMOTE来解决非平衡问题。我们用划分的训练集作为原始数据来生成SMOTE仿真数据。R语言在实际操作时，除了我们近邻参数K、随机种子外，还会遇到两个参数，决定了生成多大的SMOTE样本：perc.over = a, 决定了需要生成的少数类样本：最后少数类样本数为

$$(1+\frac{a}{100})\times N$$

N 为目前有的少数类sham样本数量；perc.under = b, 即需要从多数类样本抽样的个数：最后多数类样本数

$$(\frac{aN}{100})\times(\frac{b}{100})$$

计算调整参数，$a=2500$，$b=105$，k保持默认为5，目前的少数类样本数量$N=9$，我们有

$$(1+\frac{a}{100})\times N=(1+\frac{2500}{100})\times 9=234$$

及

$$(\frac{aN}{100})\times(\frac{b}{100})=(\frac{2500\times 9}{100})\times(\frac{105}{100})\approx 236$$

此时样本sham：CLP，即0:1的比例从9:201变为234：236，近似于1:1。

SMOTE后的仿真数据将作为新的训练集，合为470*635的矩阵。具体说明见下：

表 3：基因变量正态性检验预筛及SMOTE后数据说明表

$n=470$	变量名	详细说明	取值范围	变量数	其他
因变量	Condition	二分类数据	正常与病变	1	正常：病
自变量	NDC80等	连续型数据	≥0的实数	634	变=234:236

3.2.4 PCA与TSNE可视化对比

对420*16927的原数据，以及对根据筛除好自变量的预筛数据SMOTE生成的470*635数据，做TSNE可视化，其中正常组为绿色，病变组为黑色点：

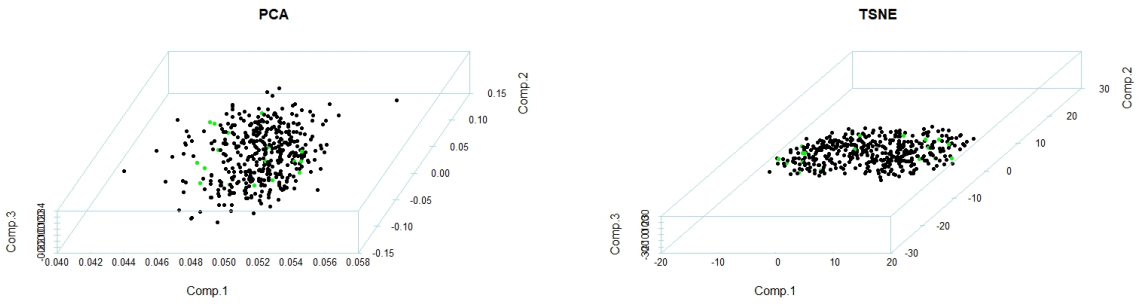

图 3-11 原始数据可视化对比

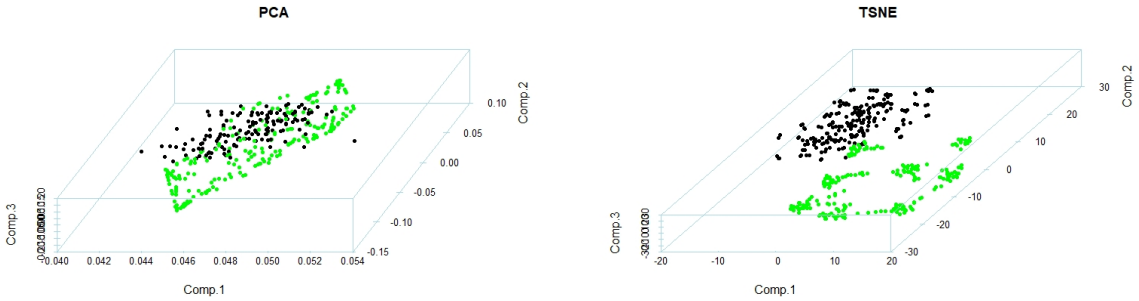

图 3-12 经基因预筛及SMOTE后的数据可视化对比

可以看到，原始高维非平衡数据无论是使用PCA还是TSNE，降维结果都非常不理想。而经过预筛和SMOTE后的生成数据经过PCA和TSNE可视化后，已经能看出正常组和病变组的二类差异，且非线性降维的TSNE区分效果明显优于线性降维的PCA，很好的满足了二分类的条件；同时，也说明了SMOTE生成新数据的合理性。

第 4 章　数据建模

在数据建模之前，有如下需要注意的地方：

（1）前文在做基因变量预筛的时候已指出，根据生物学先验知识，可知RIK基因不具有生物学意义，且数据零膨胀严重、严重不满足正态。因此，为了避免RIK类基因的加入会对回归模型（基于正态假定）产生扰动，更重要的是防止Lasso模型将该类基因筛选入最终结果（而没有实际生物学意义），我们在建模时将把所有RIK类基因从自变量数据中直接剔除，即采用基因正态性检验预筛的结果进行建模。

（2）Lasso方法要求首先对所有634个自变量数据进行标准化，因变量0-1数据不做标准化。同时，经过标准化的数据由于去除了量纲的影响，因此可按照结果的所有非零估计系数的绝对值从大到小的顺序排列，以表示系数相应变量的重要性。

（3）为防止数据随机性对结果造成的影响，本文随机种子都设置为相同种子1。

4.1　非平衡数据建模及结果

首先对进行了基因变量预筛、但未SMOTE的数据做二分类建模。采用方法综述中所述的Shrinkage+LR方法，包括MCP及Lasso族，如Elastic Net，adaptive Lasso。在建模过程中涉及到的超参数调整，本文方法叙述如下：

（1）使用Cross Validation自动选择MCP及adaptive Lasso方法的最佳压缩参数λ。

（2）对于Elastic Net，在同一个随机种子下使用网格法，在0.1-0.9中调整超参数α使得压缩后得到1-5个左右的非零系数，且AUC值最大（α越小压缩越大）。

（3）Adaptive Lasso的权重w_j由Ridge Regression得到,即$\gamma = 1$。

同时，本文将Adaptive Lasso与Elastic Net方法结合起来，以探究该合成方法是否会给结果带来更好的提升。

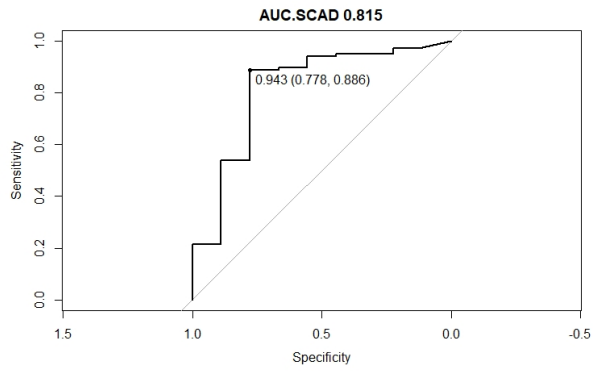

图 **4-13** MCP

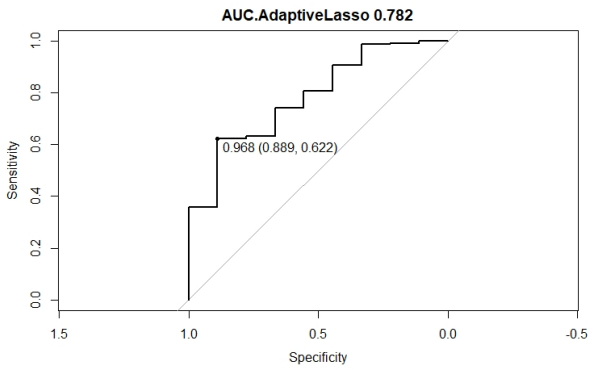

图 **4-14** Adaptive Lasso

通过R语言中的ROC包可同时画出模型在测试集测试的ROC图、AUC值和基于Yuden index给出的最优分类阈值。四个模型的ROC曲线结果如图4-13到图4-16：

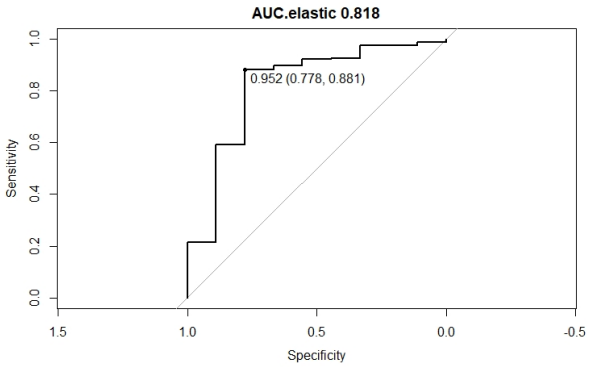

图 4-15 Elastic Net

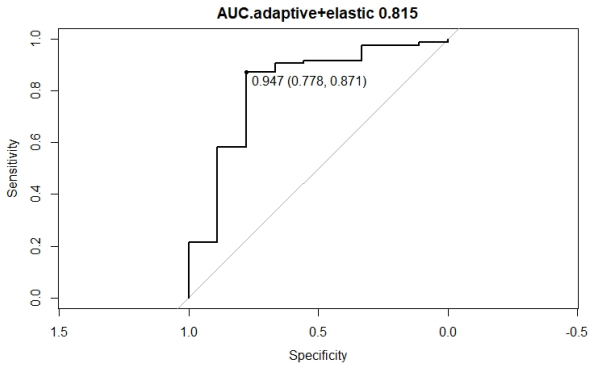

图 4-16 Adaptive Lasso and Elastic Net

由于数据已做过标准化，根据每个模型压缩出来剩余的非零系数结果，可按照系数的绝对值从大到小的顺序排列，以表示相应变量的重要性。每个模型筛选出来的基因变量名按照重要性排列顺序，结果如下：

表 4：基于训练集数据的模型筛选基因结果

模型	超参数	AUC	筛选变量（按重要性排列）	变量个数
MCP	-	0.815	Calr	1
Adaptive Lasso	$\gamma = 1$	0.782	Calr，Tspo，Pdia3，Lgals3，Serinc3，Rps2	6
Elastic Net	$\alpha = 0.4$	0.818	Calr，Tspo	2
Adaptive Lasso and Elastic Net	$\gamma = 1$, $\alpha = 0.2$	0.815	Calr，Tspo	2

可见：（1）四种方法筛选出来的关键基因个数差别不大，都在6个以内，MCP方法压缩性最大，Adaptive Lasso压缩性最小。

（2）从AUC的角度来看，Elastic Net方法的结果略胜一筹，不过总体差别不大。

（3）即便惩罚项不同，MCP的结果与其他三种Lasso及其变型方法的结果有很大共性，具有参考意义，也侧面说明了其变量筛选结果的稳健。

（4）Calr，Tspo这两个基因出现次数最多，有可能是关键基因。

最终可列入参考名单的备选关键基因有：Calr，Tspo这两个。

4.2　解决非平衡问题后的建模及结果

我们使用解决非平衡问题的SMOTE后的平衡数据进行建模，目的是为了对原基因筛选的结果做一个补充。同时也是为了从AUC结果的角度出发，评价分类问题在平衡与非平衡数据上表现的差异，并希望四种分类器能在SMOTE后的平衡数据上取得更好的表现，得到较为可靠的结果，筛选关键基因。

将SMOTE生成后的数据作为新的训练集，进行数据标准化。再将四个模型在此新训练集上训练，并在原始的测试集（经过基因预筛但非SMOTE的测试集数据）上进行测试。

模型超参数的调整方式同上。

四个模型的ROC曲线结果如下图4-17到图4-20：

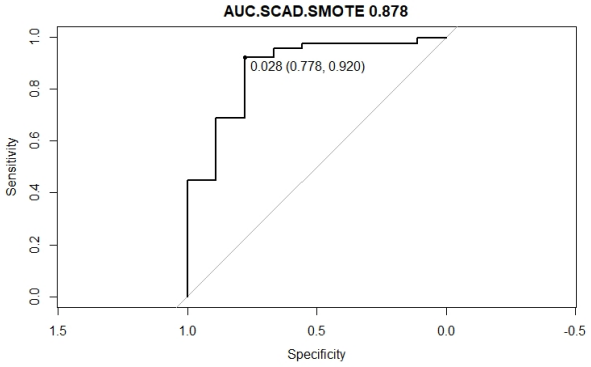

图 4-17 MCP

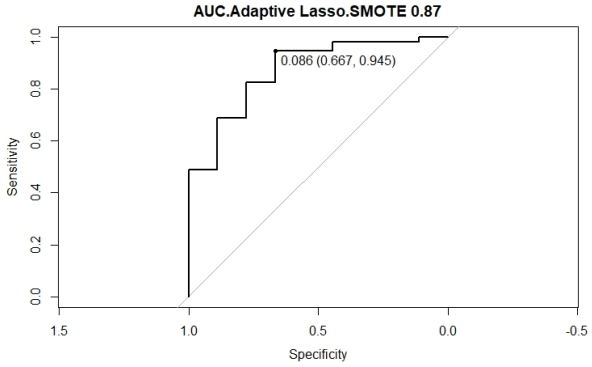

图 4-18 Adaptive Lasso

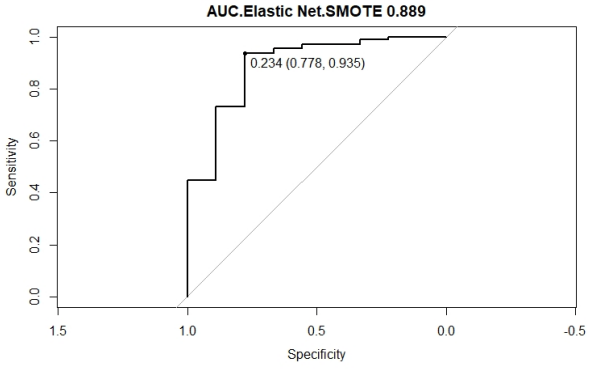

图 4-19 Elastic Net

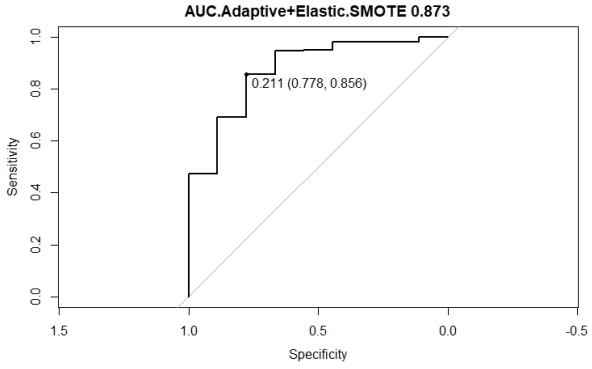

图 4-20 Adaptive Lasso and Elastic Net

同理，由于数据已做过标准化，根据每个模型压缩出来剩余的非零系数结果，可按照系数的绝对值从大到小的顺序排列，以表示相应变量的重要性。每个模型筛选出来的基因变量名按照重要性排列顺序，结果如下：

表 5：基于SMOTE后平衡训练集的模型筛选基因结果

模型	超参数	AUC	筛选变量（按重要性排列）	变量个数
MCP	-	0.878	Calr，Ubc，Naca，Sp3，Dock2，Asprv1等	8
Adaptive Lasso	$\gamma = 1$	0.870	Calr，Serinc3，Rasgrp4，Naca，Hsp90ab1，Sf3b2，Nt5c3等	21
Elastic Net	$\alpha = 0.2$	0.889	Serinc3，Rasgrp4，Naca，Hsp90ab1，Nt5c3，Sf3b2，Lilrb4a等	26
Adaptive Lasso and Elastic Net	$\gamma = 1$, $\alpha = 0.1$	0.873	Calr，Serinc3，Rasgrp4，Naca，Hsp90ab1，Sf3b2，Nt5c3等	22

可见：

（1）基于SMOTE后平衡训练集训练的分类器比非平衡训练集训练的分类器在AUC上取得了更好的效果。且选入了更大量的备选基因。

（2）四种方法筛选出来的关键基因个数除MCP外，差别不大，都在25个左右。MCP方法压缩性最大，Adaptive Lasso压缩性最小。

（3）从AUC的角度来看，仍然是Elastic Net方法的结果略胜一筹，但总体来说四个方法差别不大。

（4）即便惩罚项不同，MCP的结果与其他三种Lasso及其变型方法的结果仍有很大共性，具有参考意义，也侧面说明了其变量筛选结果的稳健。

（5）基于SMOTE后平衡训练集训练的分类器选入的备选关键基因中，与非平衡训练集训练的分类器的筛选结果，在出现次数上有一定重叠。这部分出现较多次的备选基因将作为重点考量。

统计系数绝对值（重要性）较高，且出现了较多次的基因作为主要考察对象。除Calr外，本次建模可新加入考量的备选基因有Ubc，Naca，Serinc3，Rasgrp4，Hsp90ab1，Sp3，Nt5c3，Sf3b2，Dock2，Lilrb4a等。

因此，结合4.1部分的结果，最终可列入参考名单的备选关键基因有：Calr，Tspo，Ubc，Naca，Serinc3，Rasgrp4，Hsp90ab1，Sp3，Nt5c3，Sf3b2，Dock2，Lilrb4a这12个。

第5章　多重假设检验

5.1　假设检验

本文的最后，将建模筛选出的12个备选关键基因Calr，Tspo，Ubc，Naca，Serinc3，Rasgrp4，Hsp90ab1，Sp3，Nt5c3，Sf3b2，Dock2，Lilrb4a，进行12次假设检验（即均值的假设检验），验证结论是基因的表达结果在均值上是否有显著差异，因此造成了细胞正常sham和病变CLP的区别。若有显著差异，继续探究是$\mu_{sham} < \mu_{CLP}$还是$\mu_{sham} > \mu_{CLP}$。

对于模型筛选出的基因自变量的系数，除绝对值大小排序可体现对应基因的重要性程度外，可以猜测其符号也可以体现该对应基因对细胞保持正常还是病变的影响：

（1）若符号为正，假设基因表达数据每提升一个单位，则该基因表达数据的提升会导致病变（CLP，设为变量1）概率的上升，即越有可能病变。所以从均值上看，可以猜测$\mu_{sham} < \mu_{CLP}$；

（2）若符号为负则相反，基因表达数据的下降会导致病变概率的上升，因此从均值上看，可以猜测$\mu_{sham} > \mu_{CLP}$。

表 6：12个基因系数对应符号（每个系数符号结果在8个分类器上皆保持一致）

基因	Calr	Tspo	Ubc	Naca	Serinc3	Rasgrp4
参数符号	+	+	-	-	-	-
基因	Hsp90ab1	Sp3	Nt5c3	Sf3b2	Dock2	Lilrb4a
参数符号	+	+	+	-	-	+

我们用假设检验来验证我们的猜想。本文使用的检验是T-均值检验。12次假设检验的P值分别如下：

可以看到，Sp3，Sf3b2，Dock2这3个基因，未能以小概率拒绝任意一个原假设，即认为对应细胞的基因表达结果，在正常与病变两组均值上没有显著差异。

而Calr，Tspo，Hsp90ab1，Nt5c3，Lilrb4a这5个基因，能以小概率拒绝原假设$\mu_0 \geq \mu_1$，即接受备择假设$\mu_0 < \mu_1$，认为该类基因在正常细胞的表达数据均值要显著低于病变细胞，因此造成了细胞病变；该结论与猜想吻合，即它们对应的参数符号皆为正。

另一方面，Ubc，Naca，Serinc3，Rasgrp4，这四个基因则反之，能以小概率拒绝原假设$\mu_0 \leq \mu_1$，即接受备择假设$\mu_0 > \mu_1$，认为该类基因在正常细胞的表达数据均值要显

表 7： 假设检验P值结果表

备择假设	Calr	Tspo	Ubc	Naca	Serinc3	Rasgrp4
$\mu_0 < \mu_1$	2.878e-11	2.93e-05	-	-	-	-
$\mu_0 > \mu_1$	-	-	0.008	4.885e-04	0.001	7.729e-03
$\mu_0 \neq \mu_1$	-	-	-	-	-	-
备择假设	Hsp90ab1	Sp3	Nt5c3	Sf3b2	Dock2	Lilrb4a
$\mu_0 < \mu_1$	5.709e-05	**0.346**	2.076e-04	0.747	0.852	7.033e-05
$\mu_0 > \mu_1$	-	0.654	-	**0.254**	**0.148**	-
$\mu_0 \neq \mu_1$	-	0.693	-	0.507	0.296	-

著高于病变细胞，造成细胞病变。该结论与猜想吻合，即它们对应的参数符号皆为负。

综上，最后选择出的关键基因可分为两组，$\mu_{sham} < \mu_{CLP}$组：Calr，Tspo，Hsp90ab1，Nt5c3，Lilrb4a；和$\mu_{sham} > \mu_{CLP}$组：Ubc，Naca，Serinc3，Rasgrp4。

5.2 多重假设检验

因12次假设检验中的P值结果仍有一部分基因P值在10%的显著性附近徘徊，因此本文最后再采用了多重假设检验FPR方法，对前文$\mu_{sham} < \mu_{CLP}$组与$\mu_{sham} > \mu_{CLP}$组基因P值进行修正，作为二次验证。特别的，由于假设检验中结论不显著的基因Sp3，在三组假设检验里最小P值0.346对应的备择假设为$\mu_0 < \mu_1$，且Sp3符号为正，因此结合猜想，将其归为$\mu_{sham} < \mu_{CLP}$组。同理，基因Sf3b2，Dock2归为$\mu_{sham} > \mu_{CLP}$组。

多重假设检验对原P值的修正结果如下。可见，Sp3等基因仍不显著，结果不变。

表 8： 多重假设检验Adjusted-P值结果表

备择假设	Calr	Tspo	Ubc	Naca	Serinc3	Rasgrp4
$\mu_0 < \mu_1$组	1.727e-10	8.789e-05	-	-	-	-
$\mu_0 > \mu_1$ 组	-	-	0.013	0.003	0.003	0.013
备择假设	Hsp90ab1	Sp3	Nt5c3	Sf3b2	Dock2	Lilrb4a
$\mu_0 < \mu_1$ 组	1.055e-04	**0.346**	2.491e-04	-	-	1.055e-04
$\mu_0 > \mu_1$ 组	-	-	-	**0.254**	**0.178**	-

第6章 总结

基于单细胞RNA基因测序数据，我们能够从中得到什么？以该类型为典型的生物统计类数据本身有何特点？我们是否能成功分类出正常细胞和各种疾病状态？我们能否成功找出有显著影响的基因？对于基因问题的研究，重点就是对于高维数据的特征选择问题，这也是这篇文章最为关键的地方。本文基于以上出发点，对病变细胞的分类及显著影响基因进行数据分析，从数据角度为单细胞基因测序问题提供参考，并能对结果进行初步解释。

6.1 基因变量总结

基于Shrinkage+LR的方法筛选出的12个备选基因，以及基于多重假设检验方法验证筛选出的9个最终关键基因，我们有了以下关键基因的最终结果：

表 9：最终结果

	基因	变量个数
$\mu_{sham} < \mu_{CLP}$组	Calr，Tspo，Hsp90ab1，Nt5c3，Lilrb4a	5
$\mu_{sham} > \mu_{CLP}$组	Ubc，Naca，Serinc3，Rasgrp4	4

本文从Gene Card官网搜寻和上述关键基因有关的资料。后续还得多和生化老师沟通学习。以重点出现的Calr, Ubc, Serinc3为例，举例如下：

（1）Calr (Calreticulin)：钙网蛋白编译基因。CALR突变阳性检测意味着该患者可能患有骨髓增生性肿瘤（MPN），特别是原发性血小板增多症（ET）或原发性骨髓纤维化（PMF）。从2005年开始，MPNs基因突变的发现标志着治疗MPNs的重大进展。而CALR是最近才发现的，显示出比传统MPN治疗更有效的前景，靶向治疗CALR的疗法仍在研究中。

（2）Ubc (Ubiquitin C)：泛素连接蛋白编译基因。与UBC相关的疾病包括先天性颗粒细胞肿瘤和囊性纤维化。其相关途径包括SLBP独立成熟mRNA的转运和β-肾上腺素能信号转导。

（3）Serinc3 (Serine Incorporator 3)：丝氨酸整合因子（ＳＥＲＩＮＣ）是一种膜蛋白编译基因，其家族共有 5 个成员。其中，Zhang等（2017）指出，ＳＥＲＩＮＣ５和ＳＥＲＩＮＣ３是目前最新发现的抗艾滋病病毒的天然免疫分子，可阻断病毒感染新的细胞，从而降低病毒的感染力。但对于ＳＥＲＩＮＣ家族各成员的抗病毒活性、体内

表达以及抗病毒机理，科研人员还知之甚少。

6.2 后续继续深入的思路

（1）本文的八个分类器依然属于广义线性模型的范畴。而从PCA降维与TSNE降维对比图来看，基因变量相互之间也存在关系，因此对分类的影响一定是非线性的。然而如本文开头所述，小样本的限制也不允许我们使用较为复杂的黑箱分类器，如神经网络；而对于生物统计，数据的来源限制也导致我们无法大量获取新数据。因此，如何选取合适的非线性的分类方法，使得其能在非平衡小样本上有较好的变量筛选功能，是一个值得深入思考的方向。

（2）本文所采取的分类皆基于已有的标签，因此是有监督的二分类范畴。然而事实上，在实验组402个产生了病变的未成熟中性粒细胞中，造成病变的背后模式及突变的相关关键基因有可能不同，即可能有不同种类亚型的病变产生，因此除二分类外，多分类/聚类分析也是后续研究的一个方向。对于聚类学习问题，如何确定合适的聚类数K，即多少种亚型，是一大难点。基于此问题，Kiselev等（2017）提出了新的聚类方法（同时也提供高位数据下的可视化工具），提供了一种代替多分类问题的聚类方法，通过共识方法（consensus clustering）组合多个聚类解决方案，实现了高准确性和稳健性。可作为参考。

（3）同时，本文使用的所有算法，包括线性分类器Lasso、多重假设检验等，皆建立在数据之间是iid的假设之上，这对于基因数据来说显然不合理。正因为如此，鉴定亚群、使用聚类方法更为重要。除上述提到的新聚类方法外，引入潜变量的方式也是一个思考方向。Buettner等（2015）使用引入潜变量模型的方式来解释找出了这些隐藏的亚型，成功进行了分类/聚类，可作为一个参考。

（4）由于目前仅有Lasso及其变型方法的参数估计算法，而缺乏Lasso及其变型的参数假设检验和标准误。因此，Lasso最终筛选的变量个数以及系数的参数估计大小都具有随机性，在编程中会随随机种子的不同而不同。虽然本文采取了假设检验及多重假设检验的方法进行二次验证以弥补其随机性，使得结论更加可靠稳健，但Lasso方法可能仍不是最佳的方法（如上述提到的非线性和非iid假设的原因）。除此之外，本文采用的分类器的变量筛选只实现了有或没有的功能，即只说明了系数是否非零，但未确定突变基因显著影响细胞病变时基因表达数据的关键阈值。因此，如何选取更加稳健、和能提供更多信息量的算法也是后续值得深入思考的。

（5）本文涉及到解决非平衡问题的方法采用了SMOTE方法。不过，解决非平衡的合成采样方法非常多，如V-synth合成方法等，SMOTE只是其中特别常用的一种。然而，

基于SMOTE后数据的分类器在测试集的FPR上，即原始数据的少数类分类水平上，与原始训练集的FPR结果相比，仍没有较大提高。同时，合成数据的生物学本身意义如何也暂时不清楚。可以考虑该文献提到的新方法。目前非平衡解决的问题主要针对二分类，将算法扩展为能呈现多类分类形式，是未来需深入研究的问题。

以上是针对高维非平衡分类小样本的分类/聚类存在问题的思考。对于数据本身的预处理，我们有以下两点值得思考的：

（6）解决测序数据噪声。Piccolo等(2013)使用通用表达法（UPC）来校正平台特异性背景噪音，可用于跨平台的RNA测序。

（7）前文提出，单细胞RNA-seq数据对于经典降维方法具有挑战性，因为丢失事件的普遍存在导致数据为零膨胀。我们的做法仅仅是利用正态性检验预筛丢弃掉有大量零膨胀的基因，并未做更多后续处理。而Pierson and Yau（2015）开发了一种降维方法，（Z）ero（I）nflated（F）actor（A）nalysis（ZIFA），它明确地模拟了丢失特征，并表明它提高了模拟和生物数据集的建模精度针对零膨胀问题的数据降维，可作为参考。

参考文献

[1] Benjamini, Y., & Hochberg, Y. (1995). Controlling The False Discovery Rate: A Practical And Powerful Approach To Multiple Testing. Journal of the Royal statistical society: series B (Methodological), 57(1), 289-300.

[2] Buettner, F., Natarajan, K. N., Casale, F. P., Proserpio, V., Scialdone, A., Theis, F. J., ... & Stegle, O. (2015). Computational Analysis Of Cell-To-Cell Heterogeneity In Single-Cell RNA-Sequencing Data Reveals Hidden Subpopulations Of Cells. Nature biotechnology, 33(2), 155.

[3] Chawla, N. V. , Bowyer, K. W. , Hall, L. O. , & Kegelmeyer, W. P. . (2011). Smote: Synthetic Minority Over-Sampling Technique. Journal of Artificial Intelligence Research, 16(1), 321-357.

[4] Fan, J., & Li, R. (2001). Variable Selection Via Nonconcave Penalized Likelihood And Its Oracle Properties. Journal of the American statistical Association, 96(456), 1348-1360.

[5] Kiselev, V. Y., Kirschner, K., Schaub, M. T., Andrews, T., Yiu, A., Chandra, T., ... & Hemberg, M. (2017). SC3: Consensus Clustering Of Single-Cell RNA-Seq Data. Nature methods, 14(5), 483.

[6] Maaten, L. V. D., & Hinton, G. (2008). Visualizing Data Using T-SNE. Journal of machine learning research, 9(Nov), 2579-2605.

[7] Piccolo, S. R., Withers, M. R., Francis, O. E., Bild, A. H., & Johnson, W. E. (2013). Multiplatform Single-Sample Estimates of Transcriptional Activation. Proceedings of the National Academy of Sciences, 110(44), 17778-17783.

[8] Pierson, E., & Yau, C. (2015). ZIFA: Dimensionality Reduction For Zero-Inflated Single-Cell Gene Expression Analysis. Genome biology, 16(1), 241.

[9] Tibshirani, R. J. . (1996). Regression Shrinkage and Selection Via The Lasso. Journal of the Royal Statistical Society. Series B: Methodological, 73(1), 273-282.

[10] Zhang, & Cun-Hui. (2010). Nearly Unbiased Variable Selection Under Minimax Concave Penalty. The Annals of Statistics, 38(2), 894-942.

[11] Zhang, X., Zhou, T., Yang, J., Lin, Y., Shi, J., Zhang, X., ... & Zheng, Y. H. (2017). Identification Of SERINC5-001 As The Predominant Spliced Isoform For HIV-1 Restriction. Journal of virology, 91(10), e00137-17.

[12] Zou, H. (2006). The Adaptive Lasso and Its Oracle Properties. Journal of the American statistical association, 101(476), 1418-1429.

[13] Zou, H., & Hastie, T. (2005). Regularization And Variable Selection Via The Elastic Net. Journal of the royal statistical society: series B (statistical methodology), 67(2), 301-320.

附录

```
rm(list=ls())
rawdata<-readRDS("C:/Users/EricW/Desktop/Classification based on
High-dimensional, Unbalanced RNA Data/immature_neutrophil.rds")
condition<-rawdata[['condition']]
condition<-data.frame(condition)
data<-data.frame(rawdata[['data']])
#########################################################
#转置
data<-data.frame(t(data))
condition<-condition$condition
#转置，并上factor变量
data<-cbind(condition,data)
###########
Sum<-apply(data[,-1], 1, sum)
Mean<-apply(data[,-1], 1, mean)
Std<-apply(data[,-1], 1, sd)
Median<-apply(data[,-1], 1, median)
datadiscribe<-cbind(data,Sum,Mean,Std,Median)
head(datadiscribe[,(ncol(datadiscribe)-3):ncol(datadiscribe)])
#######
sum(datadiscribe$condition=='sham')
sum(datadiscribe$condition=='CLP')
plot(Mean,Std,pch=ifelse(factor(condition)=="sham",2, 1),col=
'black',lty=1,lwd=ifelse(factor(condition)=="sham",3, 1),
ylab='Standard Deviation', main='基因表达数值均值及方差散点图')
##############缺失值
hang<-which(rowSums(is.na(data)) > 0)
hang#没有缺失值!!
#########################################################
library(glmnet)
```

```
library(ncvreg)
library(MASS)
#library(bigRR)
library(parallel)
library(caret)
library(pROC)
library(kernlab)
library(ROCR)
#################
#准备工作，拆分成训练集和测试集
set.seed(1)
Y<-data$condition
Ytestnum<-c(sample(which(Y=='sham'),length(which(Y=='sham'))/2,
replace = F),sample(which(Y=='CLP'),length(which(Y=='CLP'))/2,
replace = F))
Ytestnum<-sample(Ytestnum,length(Ytestnum),replace = F)
Ytest<-Y[Ytestnum]#三分之一分给测试集，三分之二为训练集.
Ytrain<-Y[-Ytestnum]
Y01<-rep(1,length(Y))
Y01[Y=='sham']<-0
Ytrain01<-rep(1,length(Ytrain))
Ytrain01[Ytrain=='sham']<-0 #sham的，即正常的，取0，为非常少的部分
Ytest01<-rep(1,length(Ytest))
Ytest01[Ytest=='sham']<-0
#拆分X，做一些简单的预筛
XXX<-as.matrix(data[,2:ncol(data)])
#然后取出一整列都是0的列
XX<-XXX[,-which(apply(abs(XXX),2,sum)==0)]
############################## S-W正态性检验
#先看正态性，大于0.05才认为符合正态分布!!!
P1<-rep(0,ncol(XX))
for (i in 1:ncol(XX)) {
```

```
  P1[i]<-shapiro.test(XX[,i])$p.value
}
SigP1<-sort(P1,decreasing = T)[1:635]
SigPorder1<-order(P1,decreasing = T)[1:635]
hist(XX[,SigPorder1[1]],breaks = 20)
hist(XX[,SigPorder1[635]],breaks = 20)
#####  K-S检验
P2<-rep(0,ncol(XX))
for (i in 1:ncol(XX)) {
  P2[i]<-ks.test(XX[,i],"pnorm",mean(XX[,i]), sd(XX[,i]))$p.value
}
SigP2<-sort(P2,decreasing = T)[1:635]
SigPorder2<-order(P2,decreasing = T)[1:635]
hist(XX[,SigPorder2[1]],breaks = 20)
hist(XX[,SigPorder2[635]],breaks = 20)
##########################################################
#根据正态性结果做进一步的预筛，去掉无用的基因，以免影响后续lasso
结果的筛选（特别是Rik基因）
#预筛个数n
n<-634
SigPorder3<-order(P2,decreasing = T)[1:n]
X<-XX[,SigPorder3]
Xscale<-scale(X)
Xtest<-X[Ytestnum,]
Xtrain<-X[-Ytestnum,]
Xscale<-scale(X)
Xtestscale<-Xscale[Ytestnum,]
Xtrainscale<-Xscale[-Ytestnum,]
dim(Xtrainscale)
dim(Xtestscale)
##########################################################
#[1]MCP
```

```
set.seed(1)
scad.mod<-cv.ncvreg(Xtrainscale,Ytrain01,maxit=1000000000,
family='binomial',penalty='MCP')
#plot(scad.mod)
coeffscad<-coef(scad.mod)
sum(coeffscad[-1]!=0)
pred.scad<-predict(scad.mod,Xtestscale,type='response',
family='binomial')
pred.scad2<-prediction(pred.scad,Ytest01)
perfscad<-performance(pred.scad2,'sens','spec')
performance(pred.scad2,'auc')@y.values #0.815
#0.8145384
auc.scad<-roc(Ytest01,pred.scad)
print(auc.scad)
plot(auc.scad,ylim=c(0,1),print.thres=T,main=paste
('AUC.SCAD',round(auc.scad$auc[[1]],3)))
#阈值为0.943
##############################
#[2] adaptiveLasso
set.seed(1)
ridge.mod<-cv.glmnet(Xtrainscale,Ytrain01,alpha=0,
family='binomial',type.measure = 'auc')
coeffridge<-coef(ridge.mod)
w<-1/abs(coeffridge[-1])
set.seed(1)
adalasso.mod<-cv.glmnet(Xtrainscale,Ytrain01,alpha=1,
penalty.factor=w,family='binomial',type.measure = 'auc')
coeffada2<-coef(adalasso.mod)
length(w)
sum(coeffada2[-1]!=0)
pred.ada<-predict(adalasso.mod,Xtestscale,type='response',
family='binomial')
```

```
pred.ada2<-prediction(pred.ada,Ytest01)
perfada<-performance(pred.ada2,'sens','spec')
performance(pred.ada2,'auc')@y.values#0.782
auc.ada<-roc(Ytest01,pred.ada)
print(auc.ada)
plot(auc.ada,ylim=c(0,1),print.thres=T,main=paste('AUC.ADA',
round(auc.ada$auc[[1]],3)))
#阈值为0.968
##############################
#[3]elastic net
set.seed(1)
elastic.mod<-cv.glmnet(Xtrainscale,Ytrain01,maxit=1000000,
alpha=0.6,family='binomial',type.measure = 'auc')
coeffela<-coef(elastic.mod)
sum(coeffela[-1]!=0)
pred.elas<-predict(elastic.mod,Xtestscale,type='response'
,family='binomial')
pred.elas2<-prediction(pred.elas,Ytest01)
perfelas<-performance(pred.elas2,'sens','spec')
performance(pred.elas2,'auc')@y.values#AUC达到0.818
auc.elas<-roc(Ytest01,pred.elas)
print(auc.elas)
plot(auc.elas,ylim=c(0,1),print.thres=T,main=paste
('AUC.elastic',round(auc.elas$auc[[1]],3)))
#阈值为0.952
#这里找出0.6的时候最好
for (i in seq(0,1,0.1)) {
  set.seed(1)
  elastic.mod<-cv.glmnet(Xtrainscale,Ytrain01,maxit=1000000
  ,alpha=i,family='binomial',type.measure = 'auc')
  coeffela<-coef(elastic.mod)
  print(i)
```

```
  print(sum(coeffela[-1]!=0))
  pred.elas<-predict(elastic.mod,Xtestscale,type='response'
  ,family='binomial')
  pred.elas2<-prediction(pred.elas,Ytest01)
  perfelas<-performance(pred.elas2,'sens','spec')
  print(performance(pred.elas2,'auc')@y.values)
  print('--------------------')
}
##############################
#[4]AdaptiveLasso + elastic net
set.seed(1)
elastic.mod2<-cv.glmnet(Xtrainscale,Ytrain01,maxit=1000000,
alpha=0.8,penalty.factor=w,family='binomial',type.measure = 'auc')
coeffela2<-coef(elastic.mod2)
sum(coeffela2[-1]!=0)
pred.elas<-predict(elastic.mod2,Xtestscale,type='response'
,family='binomial')
pred.elas2<-prediction(pred.elas,Ytest01)
perfelas<-performance(pred.elas2,'sens','spec')
performance(pred.elas2,'auc')@y.values#AUC达到0.815
auc.elas<-roc(Ytest01,pred.elas)
print(auc.elas)
plot(auc.elas,ylim=c(0,1),print.thres=T,main=paste('AUC.elastic
',round(auc.elas$auc[[1]],3)))
#阈值为0.947
#这里找出0.8的时候最好
for (i in seq(0,1,0.1)) {
  set.seed(1)
  elastic.mod<-cv.glmnet(Xtrainscale,Ytrain01,maxit=1000000,
  alpha=i,penalty.factor=w,family='binomial',type.measure = 'auc')
  coeffela<-coef(elastic.mod)
  print(i)
```

```
  print(sum(coeffela[-1]!=0))
  pred.elas<-predict(elastic.mod,Xtestscale,type='response'
  ,family='binomial')
  pred.elas2<-prediction(pred.elas,Ytest01)
  perfelas<-performance(pred.elas2,'sens','spec')
  print(performance(pred.elas2,'auc')@y.values)
  print('--------------------')
}
############################
Ytrain01<-as.factor(Ytrain01)
Ytest01<-as.factor(Ytest01)
train<-cbind(Ytrain01,data.frame(Xtrain))
test<-cbind(Ytest01,data.frame(Xtest))
##########################################
#先试试TSNE能不能看出啥—
library(Rtsne)
library(rgl)
library(scatterplot3d)
color<-rep('black', length(Y))
color[which(Y=='sham')]<-'green'#绿色是sham!!!!!!!!!!!!
set.seed(1)
tsne2<-Rtsne(X,dims = 2)
plot(tsne2$Y[,1],tsne2$Y[,2],main='TSNE',col=color,pch=20)
set.seed(1)
tsne3<-Rtsne(X,dims = 3,perplexity = 30,verbose = T,
max_iter = 1000)
scatterplot3d(tsne3$Y[,1],tsne3$Y[,2],tsne3$Y[,3],color=color,
              highlight.3d=F,pch = 20,angle=80,main='TSNE',
              type = 'p',grid = F,col.grid = "lightblue",
              col.axis = 'lightblue',col.lab = T,
              scale.y =2,xlim = c(-20,20),ylim = c(-25,25),
              zlim = c(-30,30))
```

```
#画PCA
df.pr <- princomp(t(X))
plot(df.pr$loadings[,1],df.pr$loadings[,2],main='PCA',
col=color,pch=20,xlab = 'Comp.1',ylab = 'Comp.2')
scatterplot3d(df.pr$loadings[,1],df.pr$loadings[,2],
df.pr$loadings[,3],color=color,
highlight.3d=F,pch = 20,angle=80,main='PCA',
type = 'p',grid = F,col.grid = "lightblue",
col.axis = 'lightblue',col.lab = T,
scale.y =5,xlab = 'Comp.1',ylab = 'Comp.2',zlab = 'Comp.3')

##################
#SMOTE
library(DMwR)
#####################################################
#now using SMOTE to create a more "balanced problem"
set.seed(1)
dataSMOTE <- SMOTE(Ytrain01 ~ ., train,perc.over = 2500
,perc.under = 105,k=5)
table(dataSMOTE$Ytrain01)
dim(dataSMOTE)#过采样后，变成了470个数!
YSMOTE<-dataSMOTE$Ytrain01
Y01.SMOTE<-rep(0,length(YSMOTE))
Y01.SMOTE[YSMOTE=='1']<-1  #0是sham!!!
X.SMOTE<-dataSMOTE[,2:ncol(dataSMOTE)]
Xscale.SMOTE<-scale(X.SMOTE)
Ytrain01.SMOTE<-Y01.SMOTE
Xtrain.SMOTE<-X.SMOTE
Xtrainscale.SMOTE<-Xscale.SMOTE
dim(Xtrainscale.SMOTE)
#########
#SMOTE后的数据TSNE
```

```
color.SMOTE<-rep('black', length(YSMOTE))
color.SMOTE[which(YSMOTE=='0')]<-'green'  #绿色是sham!!!!!!!
set.seed(1)
tsne2<-Rtsne(X,dims = 2)
plot(tsne2$Y[,1],tsne2$Y[,2],main='TSNE',col=color.SMOTE,pch=20)
set.seed(1)
tsne.SMOTE<-Rtsne(X.SMOTE,dims = 3,perplexity = 30,
check_duplicates = FALSE,verbose=T,max_iter=1000)
scatterplot3d(tsne.SMOTE$Y[,1],tsne.SMOTE$Y[,2],
tsne.SMOTE$Y[,3],color=color.SMOTE,
              highlight.3d=F,pch = 20,angle=80,main='TSNE',
              type = 'p',grid = F,col.grid = "lightblue",
              col.axis = 'lightblue',col.lab = T,
              scale.y =5,xlim = c(-20,20),ylim = c(-25,25)
              ,zlim = c(-30,30))
#画PCA
set.seed(1)
df.pr <- princomp(t(X.SMOTE))
plot(df.pr$loadings[,1],df.pr$loadings[,2],main='PCA',
col=color.SMOTE,pch=20,xlab = 'Comp.1',ylab = 'Comp.2')
scatterplot3d(df.pr$loadings[,1],df.pr$loadings[,2],
df.pr$loadings[,3],color=color.SMOTE,
highlight.3d=F,pch = 20,angle=80,main='PCA',
type = 'p',grid = F,col.grid = "lightblue",
col.axis = 'lightblue',col.lab = T,
scale.y =5,xlab = 'Comp.1',ylab = 'Comp.2',zlab = 'Comp.3')
##########################################
#SMOTE后数据建模
#1.MCP
set.seed(1)
scad.mod.SMOTE<-cv.ncvreg(Xtrainscale.SMOTE,
Ytrain01.SMOTE,family='binomial',penalty='SCAD')
```

```
coeffscad.SMOTE<-coef(scad.mod.SMOTE)
sum(coeffscad.SMOTE[-1]!=0)
pred.scad.SMOTE<-predict(scad.mod.SMOTE,Xtestscale,
type='response',family='binomial')
pred.scad.SMOTE2<-prediction(pred.scad.SMOTE,Ytest01)#
perfscad.SMOTE<-performance(pred.scad.SMOTE2,'sens','spec')
performance(pred.scad.SMOTE2,'auc')@y.values#0.8783858
auc.scad.SMOTE<-roc(Ytest01,pred.scad.SMOTE)
print(auc.scad.SMOTE)
plot(auc.scad.SMOTE,ylim=c(0,1),print.thres=T,main=paste
('AUC.SCAD.SMOTE',round(auc.scad.SMOTE$auc[[1]],3)))
##########################################
#2.Adaptive
set.seed(1)
ridge.mod2.SMOTE<-cv.glmnet(Xtrainscale.SMOTE,Ytrain01.SMOTE,
alpha=0,family='binomial',type.measure = 'auc')
w.SMOTE<-1/abs(coef(ridge.mod2.SMOTE)[-1])
set.seed(1)
elastic.mod.SMOTE<-cv.glmnet(Xtrainscale.SMOTE,Ytrain01.SMOTE
,alpha=1,penalty.factor=w.SMOTE,family='binomial',
type.measure = 'auc')
coeffela.SMOTE<-coef(elastic.mod.SMOTE)
sum(coeffela.SMOTE[-1]!=0)
pred.elas.SMOTE<-predict(elastic.mod.SMOTE,Xtestscale,
type='response',family='binomial')
pred.elas2.SMOTE<-prediction(pred.elas.SMOTE,Ytest01)
perfelas.SMOTE<-performance(pred.elas2.SMOTE,'sens','spec')
performance(pred.elas2.SMOTE,'auc')@y.values#0.870094
auc.elas.SMOTE<-roc(Ytest01,pred.elas.SMOTE)
print(auc.elas.SMOTE)
plot(auc.elas.SMOTE,ylim=c(0,1),print.thres=T,main=paste
('AUC.ELAS.SMOTE',round(auc.elas.SMOTE$auc[[1]],3)))
```

```
##############################################
#3.elastic
set.seed(1)
elastic.mod.SMOTE<-cv.glmnet(Xtrainscale.SMOTE,Ytrain01.
SMOTE,alpha=0.8,family='binomial',type.measure = 'auc')
coeffela.SMOTE<-coef(elastic.mod.SMOTE)
sum(coeffela.SMOTE[-1]!=0)
pred.elas.SMOTE<-predict(elastic.mod.SMOTE,Xtestscale,
type='response',family='binomial')
pred.elas2.SMOTE<-prediction(pred.elas.SMOTE,Ytest01)
perfelas.SMOTE<-performance(pred.elas2.SMOTE,'sens','spec')
performance(pred.elas2.SMOTE,'auc')@y.values#0.8888889
auc.elas.SMOTE<-roc(Ytest01,pred.elas.SMOTE)
print(auc.elas.SMOTE)
plot(auc.elas.SMOTE,ylim=c(0,1),print.thres=T,main=paste
('AUC.ELAS.SMOTE',round(auc.elas.SMOTE$auc[[1]],3)))
#选出0.8
for (i in seq(0,1,0.1)) {
  set.seed(1)
  elastic.mod.SMOTE<-cv.glmnet(Xtrainscale.SMOTE,Ytrain01
  .SMOTE,alpha=i,family='binomial',type.measure = 'auc')
  coeffela.SMOTE<-coef(elastic.mod.SMOTE)
  print(i)
  print(sum(coeffela.SMOTE[-1]!=0))
  pred.elas.SMOTE<-predict(elastic.mod.SMOTE,Xtestscale
  ,type='response',family='binomial')
  pred.elas.SMOTE2<-prediction(pred.elas.SMOTE,Ytest01)
  perfelas<-performance(pred.elas.SMOTE2,'sens','spec')
  print(performance(pred.elas.SMOTE2,'auc')@y.values)
  print('--------------------')
}
##############################################
```

```
#4.adaptiveLasso + elastic net
set.seed(1)
ridge.mod2.SMOTE<-cv.glmnet(Xtrainscale.SMOTE,Ytrain01
.SMOTE,alpha=0,family='binomial',type.measure = 'auc')
w.SMOTE<-1/abs(coef(ridge.mod2.SMOTE)[-1])
set.seed(1)
elastic.mod.SMOTE<-cv.glmnet(Xtrainscale.SMOTE,Ytrain01
.SMOTE,alpha=0.9,penalty.factor=w.SMOTE,family='binomial',
type.measure = 'auc')
coeffela.SMOTE<-coef(elastic.mod.SMOTE)
sum(coeffela.SMOTE[-1]!=0)
pred.elas.SMOTE<-predict(elastic.mod.SMOTE,Xtestscale,type='
response',family='binomial')
pred.elas2.SMOTE<-prediction(pred.elas.SMOTE,Ytest01)
perfelas.SMOTE<-performance(pred.elas2.SMOTE,'sens','spec')
performance(pred.elas2.SMOTE,'auc')@y.values#0.873
auc.elas.SMOTE<-roc(Ytest01,pred.elas.SMOTE)
print(auc.elas.SMOTE)
plot(auc.elas.SMOTE,ylim=c(0,1),print.thres=T,main=paste('
AUC.ELAS.SMOTE',round(auc.elas.SMOTE$auc[[1]],3)))
#这里找出0.9的时候最好
set.seed(1)
ridge.mod2.SMOTE<-cv.glmnet(Xtrainscale.SMOTE,Ytrain01.SMOTE,
alpha=0,family='binomial',type.measure = 'auc')
w.SMOTE<-1/abs(coef(ridge.mod2.SMOTE)[-1])
for (i in seq(0,1,0.1)) {
  set.seed(1)
  elastic.mod.SMOTE<-cv.glmnet(Xtrainscale.SMOTE,Ytrain01.SMOTE
  ,alpha=i,penalty.factor=w.SMOTE,family='binomial',
  type.measure = 'auc')
  coeffela.SMOTE<-coef(elastic.mod.SMOTE)
  print(i)
```

```
  print(sum(coeffela.SMOTE[-1]!=0))
  pred.elas.SMOTE<-predict(elastic.mod.SMOTE,Xtestscale,
  type='response',family='binomial')
  pred.elas.SMOTE2<-prediction(pred.elas.SMOTE,Ytest01)
  perfelas<-performance(pred.elas.SMOTE2,'sens','spec')
  print(performance(pred.elas.SMOTE2,'auc')@y.values)
  print('--------------------')
}
############################################################
#验证这12个基因:
#Calr, Tspo, Ubc, Naca, Serinc3, Rasgrp4, Hsp90ab1, Sp3, Nt5c3, Sf3b2,
Dock2, Lilrb4a
#1.######################
Calr0<-X[,'Calr'][which(Y01==0)]#取出是sham的, 即正常的
Calr1<-X[,'Calr'][which(Y01==1)]#取出不正常的CLP部分
t.test(Calr0,Calr1,var.equal = TRUE,alternative = "less")#sham显
著＜CLP
pCalr<-t.test(Calr0,Calr1,var.equal = TRUE,alternative
= "less")$p.value
#2.######################
Tspo0<-X[,'Tspo'][which(Y01==0)]#取出是sham的, 即正常的
Tspo1<-X[,'Tspo'][which(Y01==1)]#取出不正常的CLP部分
t.test(Tspo0,Tspo1,var.equal = TRUE,alternative = "less")#sham显
著＜CLP
pTspo<-t.test(Tspo0,Tspo1,var.equal = TRUE,alternative
 = "less")$p.value
#3.######################
Ubc0<-X[,'Ubc'][which(Y01==0)]#取出是sham的, 即正常的
Ubc1<-X[,'Ubc'][which(Y01==1)]#取出不正常的CLP部分
t.test(Ubc0,Ubc1,var.equal = TRUE,alternative = "greater")#sham显
著＞CLP
pUbc<-t.test(Ubc0,Ubc1,var.equal = TRUE,alternative
```

```
= "greater")$p.value
#4.####################
Naca0<-X[,'Naca'][which(Y01==0)]#取出是sham的，即正常的
Naca1<-X[,'Naca'][which(Y01==1)]#取出不正常的CLP部分
t.test(Naca0,Naca1,var.equal = TRUE,alternative = "greater")#sham显
著＞CLP
pNaca<-t.test(Naca0,Naca1,var.equal = TRUE,alternative
 = "greater")$p.value
#5.####################
Serinc30<-X[,'Serinc3'][which(Y01==0)]#取出是sham的，即正常的
Serinc31<-X[,'Serinc3'][which(Y01==1)]#取出不正常的CLP部分
t.test(Serinc30,Serinc31,var.equal = TRUE,alternative = "greater")
#sham显著＞CLP
pSerinc3<-t.test(Serinc30,Serinc31,var.equal = TRUE,alternative
= "greater")$p.value
#6.####################
Rasgrp40<-X[,'Rasgrp4'][which(Y01==0)]#取出是sham的，即正常的
Rasgrp41<-X[,'Rasgrp4'][which(Y01==1)]#取出不正常的CLP部分
t.test(Rasgrp40,Rasgrp41,var.equal = TRUE,alternative = "greater")
#sham显著＞CLP
pRasgrp4<-t.test(Rasgrp40,Rasgrp41,var.equal = TRUE,alternative
 = "greater")$p.value
#7.####################
Hsp90ab10<-X[,'Hsp90ab1'][which(Y01==0)]#取出是sham的，即正常的
Hsp90ab11<-X[,'Hsp90ab1'][which(Y01==1)]#取出不正常的CLP部分
t.test(Hsp90ab10,Hsp90ab11,var.equal = TRUE,alternative = "less")
#sham显著＜CLP
pHsp90ab1<-t.test(Hsp90ab10,Hsp90ab11,var.equal = TRUE,alternative
 = "less")$p.value
#8.####################
Sp30<-X[,'Sp3'][which(Y01==0)]#取出是sham的，即正常的
Sp31<-X[,'Sp3'][which(Y01==1)]#取出不正常的CLP部分
```

```
t.test(Sp30,Sp31,var.equal = TRUE,alternative = "two.sided")#不显著不同
t.test(Sp30,Sp31,var.equal = TRUE,alternative = "less")#不显著不同
t.test(Sp30,Sp31,var.equal = TRUE,alternative = "greater")#不显著不同
pSp3<-t.test(Sp30,Sp31,var.equal = TRUE,alternative = "less")
$p.value
#9.#####################
Nt5c30<-X[,'Nt5c3'][which(Y01==0)]#取出是sham的，即正常的
Nt5c31<-X[,'Nt5c3'][which(Y01==1)]#取出不正常的CLP部分
t.test(Nt5c30,Nt5c31,var.equal = TRUE,alternative = "less")#sham显著＜CLP
pNt5c3<-t.test(Nt5c30,Nt5c31,var.equal = TRUE,alternative
 = "less")$p.value
#10.#####################
Sf3b20<-X[,'Sf3b2'][which(Y01==0)]#取出是sham的，即正常的
Sf3b21<-X[,'Sf3b2'][which(Y01==1)]#取出不正常的CLP部分
t.test(Sf3b20,Sf3b21,var.equal = TRUE,alternative = "two.sided")
t.test(Sf3b20,Sf3b21,var.equal = TRUE,alternative = "less")
t.test(Sf3b20,Sf3b21,var.equal = TRUE,alternative = "greater")
pSf3b2<-t.test(Sf3b20,Sf3b21,var.equal = TRUE,alternative
 = "greater")$p.value
#11.#####################
Dock20<-X[,'Dock2'][which(Y01==0)]#取出是sham的，即正常的
Dock21<-X[,'Dock2'][which(Y01==1)]#取出不正常的CLP部分
t.test(Dock20,Dock21,var.equal = TRUE,alternative = "two.sided")
t.test(Dock20,Dock21,var.equal = TRUE,alternative = "less")
t.test(Dock20,Dock21,var.equal = TRUE,alternative = "greater")
pDock2<-t.test(Dock20,Dock21,var.equal = TRUE,alternative
= "greater")$p.value
#12.##############################
```

```
Lilrb4a0<-X[,'Lilrb4a'][which(Y01==0)]#取出是sham的，即正常的
Lilrb4a1<-X[,'Lilrb4a'][which(Y01==1)]#取出不正常的CLP部分
t.test(Lilrb4a0,Lilrb4a1,var.equal = TRUE,alternative = "less")
#sham显著＜CLP
pLilrb4a<-t.test(Lilrb4a0,Lilrb4a1,var.equal = TRUE,alternative
 = "less")$p.valu
##################################
#FDR
Pless<-cbind(pCalr,pTspo,pHsp90ab1,pNt5c3,pLilrb4a,pSp3)
Pgreater<-c(pUbc,pNaca,pSerinc3,pRasgrp4,pSf3b2,pDock2)
PFDRless<-p.adjust(Pless,method = 'fdr',n=length(Pless))
PFDRgreater<-p.adjust(Pgreater,method = 'fdr',n=length(Pgreater))
```