基于贝叶斯算法的等位基因特异的 m^{6}A 鉴定算法开发

【摘要】

m^{6}A 是最为普遍的 mRNA 甲基化修饰，多项研究表明它调控着我们身体内的许多重要的代谢过程，与肥胖、癌症以及其他的人类疾病有密切的联系。而随着功能基因组学的迅速发展，越来越多的研究表明非编码修饰环节存在等位基因特异的偏好性，对于性状变异存在影响。因此，判断 m^{6}A 是否具有等位基因特异性能够有效地帮助我们理解它的功能及机理，具有十分重要的意义。而引入 SNP 位点进行等位基因特异性的判断已在多项研究中实现，具有良好效果。因此，本文使用生物信息技术，结合数理统计学方法，构建出模型对实验数据进行分析，以判断 m^{6}A 位点是否具有等位基因特异性。我们使用模型，对两组通过免疫沉淀法获得的 m^{6}A 高通量测序数据进行分析，鉴定出 9 个具有等位基因特异性的 m^{6}A 位点。

【关键词】: m^{6}A；等位基因特异性；贝叶斯统计；MCMC

The development of m^{6}A site's allele specificity determination algorithm based on bayesian algorithm

[ABSTRACT]

Methylation of the N^6 position of adenosine is one of the most common mRNA methylation modification. Studies have shown that it controls many important metabolic process in our body, and has strong influence in obesity, cancer and other human diseases. Meanwhile, with the rapid development of functional genomics, more and more studies have shown that many non-coding modification process is allele specific, which leads to trait variation. Therefore, finding out whether it is allele specific is important for us to understand its function. By considering SNP site, we used the biological information technology, which combined with the mathematical statistical methods, to build a model in order to find out whether the m^{6}A allele specific by using high-throughput sequencing data obtained from immune precipitation. By using the model to analyze two groups of high-throughput sequencing data obtained by immune precipitation, we identified 11 m^{6}A sites which are allele specific.

[Keywords]: m^{6}A, alleles specific, bayesian algorithm, MCMC

目录

1. 前言

至今为止已知的RNA化学修饰有100余种，它们发生在包括rRNA，tRNA，mRNA等各种类型的RNA中，调控着我们身体内的各种代谢过程，具有重要意义。在RNA的化学修饰中，大部分为甲基化修饰。发生在RNA的N^6-甲基腺苷上的甲基化修饰m^6A（Methylation of the N^6 position of adenosine ）是最重要的甲基化修饰之一，受到广泛的关注。它调控着我们身体内的许多重要的代谢过程，与肥胖、癌症以及其他的人类疾病有密切的联系。[1-3]

由于被甲基化的碱基A并没有改变它的碱基配对能力，目前还不能通过直接测序检测m^6A的位点。现在检测m^6A的主要方法为免疫沉淀法（Immunoprecipitation，IP）与数据分析的结合。首先利用免疫沉淀法获得与m^6A抗体进行孵育的RNA片段测序数据，再使用峰值算法对测序数据进行分析，获得m^6A位点所在区域，最后利用已知的m^6A位点信息构建及训练模型，并使用模型对m^6A位点进行预测。

单核苷酸多态性（SNP）是不同个体的 DNA 序列上的单个碱基的差异，它遗传稳定等优点使它备受关注，成为第三代遗传诊断的标记。[4]它在人类基因组中广泛存在，能够影响基因的表达。它在医学以及对鱼、虾的研究都作为优秀的定位标记被广泛使用。[5-7]利用 SNP 位点以及统计检验的方法，我们能够了解 m^6A 是否具有等位基因特异性，从而进行深入研究。

2. 研究意义及目的

许多表观遗传学的调控过程常常会出现等位基因特异的偏好性，解析这些等位基因特异的调控过程也是深入了解机体生命活动的切入点。许多研究表明，在生物体中的一些重要调控过程，包括 DNA 修饰等，均与有等位基因的特异性。[8]而还有一些研究表明，研究表明，受局部和遥远遗传变异影响的机制所控制的 RNA 表达过程，最终通过蛋白质数量和功能的改变引起疾病。[9,10]因此，研究 m^6A

是否具有等位基因特异性对于人们进一步了解 m^6A 及其他 RNA 转录后修饰以及研究 m^6A 对于人体内代谢活动的调解及疾病的影响都具有重要意义。

但是，被甲基化的碱基 A 并没有改变它的碱基配对能力，目前还不能通过直接测序检测 m^6A 的位点。现在，采用实验的方法很难精确地、高效地确定 m^6A 的修饰位点，因此在组学水平上解析 m^6A 修饰的过程十分困难。

同时，随着功能基因组学的迅速发展，越来越多的研究表明非编码修饰环节对于性状变异的影响。[8, 11] Asaf Hellman 等人已经证实不同的甲基化会对基因的表达有不同的影响。[12]而根据 Yingying Zhang 等人的报道，等位基因特异性甲基化可以导致甲基化基因拷贝的等位抑制。[13]此外，因此，我们好奇发生在 RNA 的 N^6-甲基腺苷上的甲基化修饰（m^6A）是否也会对 DNA 的转录、表达等过程产生影响。

由于 m^6A 的特性导致其鉴定的困难性，针对 m^6A 修饰的等位基因特异的偏好性尚未见报道。而随着我们对于细胞、基因、蛋白研究手段的精细化与准确化以及高通量测序技术的发展，生物信息学手段在细胞层面发挥着越来越重要的作用。[14, 15]最近的研究中，一种基于高通量测序的全基因组的等位基因特异的方法被报道，它允许我们直接衡量基因多态性对于基因表达以及染色质状态的影响。它具有消除对基因表达或 dna -蛋白相互作用的环境或跨作用影响，提供更高的敏感性，以揭示序列和表观遗传变异对顺式调控元件的直接影响的好处，具有重要意义。[11]SNP 位点可以帮助我们判断转录片段来源于父本还是母本，因此在判断等位基因特异性的过程中的能够发挥重要作用。在酵母中，已实现利用 SNP 位点实现等位基因特异性的判断。[16-18]因此，我们希望引入 SNP 位点，结合计算生物学手段进行辅助，在已搭建好的 SNA 鉴定流程的基础上研究转录组中等位基因特异的 m^6A 调控过程。

3. 材料与方法

本实验使用 Dan Dominissini 等人采用免疫沉淀法获得的 m^6A 数据[19]作为实验材料进行生物信息学处理，利用 R 语言中的 MeTPeak[20]包获得 m^6A 位点的峰值区域，

再通过 SRAMP[21]软件获得单碱基精度的 m^{6}A 位点，freebayes[22]软件获得 SNP 位点，最后利用获得 m^{6}A 及 SNP 位点数据，并利用所获得的数据构建检验模型。本实验中使用的所有的材料以及软件均可以在网上免费获得。

3.1 数据材料

本实验共使用来自人类细胞的两组通过免疫测序法获得的高通量数据作为材料进行 m^{6}A 位点等位基因特异性的检验。第一组数据是 Dan Dominissini 等人采用免疫沉淀法获得的[19],该材料能通过 SRA ToolKit 从 GEO accession 获得，其编号为 GSE37003。它是由 Illumina Genome Analyzer IIx 获得的高通量测序数据。其中，利用 m^{6}A 抗体在从 HepG2 细胞中通过免疫沉淀技术获得 IP 样本 4 个，以及作为对照组的 Input 样本 3 个。我们选取了 3 个 IP 样本以及 3 个 Input 样本作为三组生物学重复进行试验。

第二组数据是 Ke S 等人从人类细胞以及小鼠细胞中获得的[23],它通过 Illumina HiSeq 2000 测序获得，包括 18 个样本数据，我们从中取来自人类细胞的 3 个 Input 样本以及 3 个 IP 样本作为数据用于分析。该材料能通过 SRA ToolKit 从 GEO accession 获得，其编号为 GSE37003。我们使用的测序样本为 SRR2120887、SRR2120888、SRR2120889、SRR2120890、SRR2120891、SRR2120892。

SRA ToolKit[24]是 NCBI 提供的一个软件，它能够帮助我们以编程的方式从 NCBI 数据库中获得 SRA 文件并将其转化为我们所需的格式。从 GEO accession 获得实验数据的流程如下：

1. 通过 NCBI 中将 SRA ToolKit 工具下载并安装到服务器上；
2. 使用 Prefetch --max-size 100000000 <SRA accession>将所需要的 SRA 文件下载到服务器中；
3. 使用 fastq-dump <SRA accession>命令将 SRA 文件转化为我们所需的 fastq 文件格式。

3.2 研究方法

本实验旨在构建模型用于判断 m^{6}A 位点是否具有等位基因特异性。但由于被

甲基化的碱基 A 并没有改变它的碱基配对能力，目前还不能通过直接测序检测 m^6A 的位点。m^6A 的特性导致其鉴定较为困难，因此，针对 m^6A 修饰的等位基因特异的偏好性尚未见报道。

本实验尝试借助单核苷酸多态性（single nucleotide polymorphism，SNP）达到这一目的。首先，我们能够根据实验获得的高通量测序数据通过生物信息学手段进行 m^6A 位点以及 SNP 位点的鉴定，获得基因组上的 m^6A 位点的位置以及 SNP 位点。然后，我们可以找到位于同一段测定序列（reads）上的 m^6A 位点以及 SNP 位点。由于 SNP 位点与 m^6A 位点位于同一段测定序列（reads）上，因此如果在这些 reads 上出现转录的偏好性，那么，我们就可以认为该 m^6A 位点具有等位基因特异性。对于每一组 m^6A 位点以及 SNP 位点，我们能够根据每条 reads 上 SNP 位点的碱基判断其来自母本还是来自副本。根据 reads 的总数以及来自父本及母本地数目，我们能够构建模型模拟其分布，并使用 MCMC 获得模型中的参数。

我们利用 R 语言中的 MeTPeak[20]包获得 m^6A 位点的峰值区域， SRAMP[21]软件获得单碱基精度的 m^6A 位点，再通过 freebayes[22]软件获得 SNP 位点。在获得 SNP 位点后，我们通过比较 SNP 位点与 m^6A 位点在基因组上的位置，来找到与 m^6A 位点位置相近的 SNP 位点。通过计算在 m^6A 位点旁的 SNP 位点是否具有表达的偏好性，从而判断出与它相近的 m^6A 是否具有表达的偏好性，从而判断 m^6A 是否具有等位基因特异性。具体流程图如下图所示：

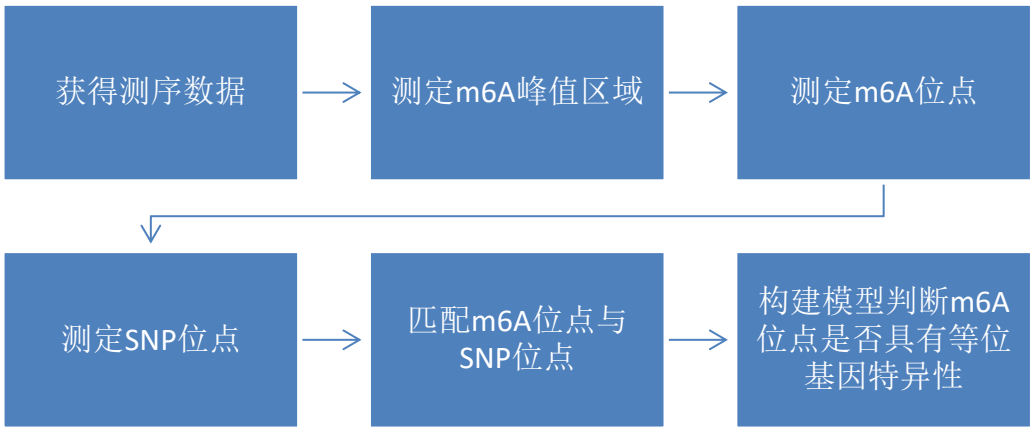

图 1 实验总流程图

Figure 1 Flow path of whole process

3.2.1 获得 m^{6}A 峰值区域

为了使后续分析中预测得的 m^{6}A 位点更为准确，我们需要先对通过免疫沉淀法获得的高通量测序数据进行分析，获得 m^{6}A 的峰值区域，即 m^{6}A 发生修饰最后可能的区域。我们共使用了两种方式对 m^{6}A 峰值区域进行判定。由于使用 R 语言中的 MeTPeak[20]包能够获得更多有效的实验数据，因此在本实验中，我们使用 R 语言中的 MeTPeak[20]包对高通量测序数据进行分析，获得 m^{6}A 峰值区域。

第一个方法基于 MACS 软件[25, 26]，它的重要步骤在于使用 MACS 基于全局定位信号富集区。它使用小范围的窗口在基因组上扫描，通过比较经过 m^{6}A 抗体处理的 IP 样本和作为对照组的 Input 样本的测定序列数（reads）的差异，利用统计学方法判定该区域是否为峰值。[27]

第二个方法是使用 MeTPeak 包获得峰值区域数据。MeTPeak 包是一种新的、基于图形的寻找峰值区域的方法，它同样是用于在转录组水平上通过甲基化免疫沉淀测序技术（Methylated RNA Immunoprecipitation Sequencing technology，MeRIP-seq）获得数据中检测 m^{6}A 位点。MeTPeak 显式地模拟了 m^6 位点的读取计数，并引入了一层 Beta 变量来获得方差和一个隐式马尔可夫模型来描述测定序列对于位点的依赖性。此外，它还包含了一种约束牛顿法以及一种对数障碍函数用于计算难以分析的、明确受约束的 beta 变量。[16, 20]

使用 MACS 获得 m^{6}A 峰值区域的流程如下：

1. 从 MACS 的安装网站[28]中根据指示在服务器上安装 MACS；
2. 从网站[29]中下载 bowtie 并根据指示在服务器上进行安装；
3. 从 github 上下载 bedtools[30, 31]并根据指示进行安装；
4. 使用 bowtie 软件将 fastq 文件定位到基因组上，获得 sam 文件；
5. 使用 MACS 获得峰顶点；
6. 筛选出错误率低于 5%的位点；
7. 使用 sort 获得位点前后 50nt 的序列作为峰值区域；
8. 使用 intersectBed 软件获得记录正负链信息的 bed 文件；
9. 使用 fastaFromBed 软件获得对应的 fasta 文件。

使用 MeTPeak 包获得 m^{6}A 峰值区域的流程如下：

1. 从网站[29]中下载 bowtie 并根据指示在服务器上安装 bowtie；
2. 根据网站[32]指示下载并安装 samtools；
3. 使用 bowtie 软件将 fastq 文件定位到基因组上，获得 sam 文件；
4. 使用 samtools 软件将 sam 文件转化为其二进制文件——bam 文件；
5. 使用 samtools 软件对 bam 文件进行排序并制作索引文件——bai 文件；
6. 从 ensemble 数据库中获得待检测序列的参考基因组注释文件——gtf 文件；
7. 通过网站[33]中的指引，加载 MeTPeak 包；
8. 使用 R 语言中的 MeTPeak 包，对经过处理后的测序数据进行分析，获得包含 m^6A 位点峰值信息的相关文件。

3.2.2 获得单碱基精度的 m^6A 位点

在获得 m^6A 的峰值区域之后，我们通过 SRAMP[21]软件进一步得到 m^6A 位点的预测位置。SRAMP 是一个哺乳动物 m^6A 位点的电脑预测软件。它使用了三个随机森林分类器，分别鉴定核苷酸序列的可能模式，K-邻近信息和位点独立的的核苷酸对光谱特性。它可以使用基因组序列或者 cDNA 序列作为输入值，无论使用那种输入值，SRAMP 都能够在交叉验证实验以及严格的独立的基准测试有杰出的表现。

SRAMP 分成网页版以及本地版两种版本。我们可以直接在网页[34]中提交数据对 m^6A 位点进行预测，也可以将 SRAMP 的本地版下载在服务器中使用。但是本地版被简化了，去掉了包括 KNN 编码分类器和 RNA 结构注释功能在内的这些耗时的部分，从而导致了使我们能够更快获得数据，但是降低了数据的准确性。由于我们需要对大量的 m^6A 位点区域进行鉴定，因此我们使用 SRAMP 本地版进行 m^6A 位点预测。使用 SRAMP 对 m^6A 位点预测的流程如下：

1. 从 SRAMP 的网站中将本地版 SRAMP 软件下载到服务器中；
2. 对由 MeTPeak 包获得的 m^6A 峰值区域数据进行处理，将其分割成以一个测序序列为一行的 bed 文件
3. 运行 runsramp.pl 脚本，对 bed 文件进行分析，获得含有 m^6A 位点预测结果的 txt 文件。

3.2.3 获得 SNP 位点

在获得预测的 m^6A 位点后，为了构建用于检验 m^6A 是否具有等位基因特异性，我们还需要借助 m^6A 位点周围的 SNP 位点用于等位基因特异性的判断，因此，我们需要获得 SNP 位点的数据。

我们使用 freebayes[22]软件来对处理后的测序数据进行分析，获得 SNP 位点信息。freebayes——贝叶斯变型检测器中运用了贝叶斯模型的变体，并加入了单倍型的使用。单倍型的使用不仅提高了 freebayes 对基因组信息的解释能力，而且对现有方法的检测性能有了很大的提高。它适用于任意的基因组结构、倍体结构、种群结构、样本数量和等位基因数量。加入对置信度的估计使得分析过程能够准确地使用实验数据，进一步提高了模型的效能。

使用 freebayes 工具获得 SNP 位点的流程如下：

1. 根据网站[35]的指引，在服务器上下载并安装 freebayes 工具；
2. 从 NCBI 获得 ensemble 数据库中下载参考基因组的 fasta 文件；
3. 使用 samtools 软件对需要进行分析的 bam 文件进行排序并制作索引文件——bai 文件；
4. 运行 freebayes -f <fasta file> <bam file> > <output file>获得结果文件——vcf 文件。

3.2.4 构建模型

在获得 m^6A 位点信息以及 SNP 位点信息后，我们需要找到 m^6A 位点附近的 SNP 位点信息，通过判断 SNP 位点是否具有表达的偏好性从而判断 m^6A 位点是否具有表达的偏好性，从而判断 m^6A 位点是否具有等位基因的特异性。构建模型判断 m^6A 位点是否具有等位基因特异性的流程如下图所示：

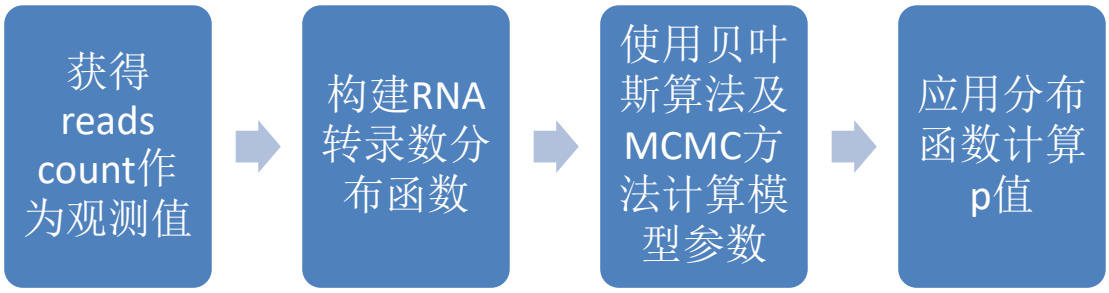

图 2 判断 m^{6}A 位点是否具有等位基因特异性流程示意图

Figure 2 Flow path for testing whether m^{6}A is allele specific

首先，我们需要提取出有效的 m^{6}A 位点以及 SNP 位点。SRAMP 对所预测出的 m^{6}A 依据置信度高低进行了评分，我们可以在 Linux 服务器中使用 awk 选取出评分为 0.557 及以上的 m^{6}A 位点作为有效位点，这些位点具有中等及以上的置信度。同时，我们对使用 freebayes 获得包含 SNP 位点的 vcf 文件进行处理，使用 awk 挑选出评分高于 20，即错误率低于 1%的 SNP 位点，视其为有效的 SNP 位点。

在挑选出有效的 m^{6}A 位点以及 SNP 位点后，我们需要对包含 m^{6}A 位点以及 SNP 位点信息的文件进行处理，提取出我们所需的信息。由于我们需要找到 m^{6}A 位点周围的 SNP 位点，并使用 SNP 位点的测序数据判断其是否具有转录的偏好性，因此，我们首先需要获得 m^{6}A 位点信息。我们使用 python 编程，从 m^{6}A 位点文件中提取出 m^{6}A 位点的信息（附录 1），包括 m^{6}A 位点所在的染色体以及在该染色体上的位置。

然后，我们从预测 SNP 位点获得的 vcf 文件中提取出 SNP 位点的位置信息，以及在该位点的总的测定序列数（total reads）以及 4 个碱基中出现频数最大的碱基所在的测定序列数（reads count）。（附录 2）其中，位置信息为 SNP 位点所在的染色体以及在该染色体上的位置。total reads 为在该 SNP 位点所获得的总的测定序列数。对于每一个与参考基因组不同的碱基，该位点为该碱基的的测定序列数为 x_i，则对于每一个 SNP 位点，reads count 为该 SNP 位点中 X_i 的最

大值。

在获得所有的所需信息后，我们需要为 m^{6}A 位点找到其周围的 SNP 位点进行模型的构建。由于测定序列的长度为 50nt，因此我们使用 python 编程，获得与 m^{6}A 位点相距 25 个 nt 距离的 SNP 位点（附录 3），假定它们位于同一测定序列中，从而确保 SNP 位点的转录偏好性 m^{6}A 位点的转录偏好性一致。对于每一组 m^{6}A 与 SNP 位点，我们将其视为一个观察结果。

在获得了与 m^{6}A 位点相匹配的 SNP 位点数据后，我们构建了一个基于贝叶斯算法的模型来对 SNP 位点的转录情况进行模拟，以判断其是否具有转录偏好性。

我们使用$y_1, y_2, y_3, \cdots, y_n$代表第一个到第 n 个观察结果中的 reads count 的数目。并假定 Y 服从以下分布。

$$\mathrm{Y} = \{y_1, y_2, y_3, \cdots, y_n\}$$

$$\begin{aligned} \mathrm{Y} \sim & \frac{\pi}{\pi+\rho+\sigma}\,\mathrm{BetaBinormial}(\mathrm{N},\alpha,\alpha) \\ & + \frac{\rho}{\pi+\rho+\sigma}\,\mathrm{BetaBinormial}(\mathrm{N},\beta,\gamma) \\ & + \frac{\sigma}{\pi+\rho+\sigma}\,Possion(\lambda) \end{aligned} \tag{1}$$

$$\pi, \rho, \sigma \sim \Gamma(\mathrm{c},\mathrm{d}) \tag{2}$$

$$\alpha \sim \Gamma(\mathrm{e},\mathrm{f}) \tag{3}$$

$$\beta \sim \mathrm{Beta}(\mathrm{g},\mathrm{h}) \tag{4}$$

$$\gamma \sim \mathrm{Beta}(\mathrm{p},\mathrm{q}) \tag{5}$$

$$\lambda \sim \Gamma(\mathrm{m},\mathrm{n}) \tag{6}$$

由于测定序列在两条染色体上转录的概率随位点的变化而变化，因此，我们

使用贝塔二项分布来模拟测定序列来源于两条染色体中的可能。其中，第一个贝塔二项分布表示测定序列中来源于两条染色体可能性相同的部分，第二个贝塔二项分布表示来源于两条染色体可能性不同的部分，即有转录具有偏好性的部分，泊松分布表示可能的误差项。$\frac{\pi}{\pi+\rho+\sigma}$、$\frac{\rho}{\pi+\rho+\sigma}$、$\frac{\sigma}{\pi+\rho+\sigma}$分别表示表示测定序列中来源于两条染色体可能性相同的部分、来源于两条染色体可能性不同的部分以及可能的误差项的比例，其总和为一。在该位点处 y 值的总体数，即总的测定序列数（total reads）。,$\alpha,\beta,\gamma,\lambda$分别用于描绘三个部分分布特征的参数。由于在生物体转录的过程中具有随机性，因此我们使用贝塔分布来模拟β,γ的变化，使β,γ的值位于 0、1 之间，使得二项分布中成功的概率接近 0 或 1，使用贝塔分布来模拟$\pi,\rho,\sigma,\alpha,\lambda$，使得$\pi,\rho,\sigma,\alpha,\lambda$的值大于 0。模拟$\pi,\rho,\sigma,\alpha,\beta,\gamma,\lambda$变化的参数取值如下：

$$\mathrm{c}=9,\mathrm{d}=1;\mathrm{e}=9,\mathrm{f}=2;\mathrm{g}=8,\mathrm{h}=1.5;\mathrm{p}=8,\mathrm{q}=1.5;\mathrm{m}=0.5,\mathrm{n}=1$$

在构建好模型后，我们需要计算出$\pi,\rho,\sigma,\alpha,\beta,\gamma,\lambda$这 7 个参数的值。用于我们已经获得了 Y 的观察值，我们可以使用贝叶斯算法，通过后验概率获得先验概率。我们得到的全条件后验概率分布如下：

$$\theta=\{\pi,\rho,\sigma,\alpha,\beta,\gamma,\lambda\} \tag{7}$$

$$\begin{aligned}\mathrm{P}(\theta|\mathrm{Y}) &= \frac{P(Y|\theta)P(\theta)}{\sum P(Y|\theta)P(\theta)} \propto P(Y|\theta)P(\theta)\\ &= P(Y|\pi,\rho,\sigma,\alpha,\beta,\gamma,\lambda)P(\pi,\rho,\sigma,\alpha,\beta,\gamma,\lambda)\\ &= P(Y|\pi,\rho,\sigma,\alpha,\beta,\gamma,\lambda)P(\pi)P(\rho)P(\sigma)P(\alpha)P(\beta)P(\gamma)P(\lambda)\end{aligned} \tag{8}$$

$$\begin{aligned}&f(x|N,\alpha,\beta)_{BetaBinormial}\\ &=\frac{\Gamma(N+1)}{\Gamma(x+1)\Gamma(N-x+1)}\frac{\Gamma(x+\alpha)\Gamma(N-x+\beta)}{\Gamma(N+\alpha+\beta)}\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\end{aligned} \tag{9}$$

$$f(x|\lambda)_{possion} = \frac{\lambda^x e^{-\lambda}}{x!} \tag{10}$$

$$f(x|\alpha,\beta)_{gamma} = \frac{\beta^\alpha x^{\alpha-1} e^{-\beta x}}{\Gamma(\alpha)} \tag{11}$$

$$f(x|\alpha,\beta)_{beta} = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\beta)} \tag{12}$$

由于 7 个参数相互独立，因此我们可以将其分解成 7 个概率密度函数的相乘。带入上面贝塔二项分布、贝塔分布、伽马分布、泊松分布的概率密度函数，我们可以获得 7 个参数所服从的分布：

为了便于计算，我们将等式两边分别取对数，获得如下等式：

$$\begin{aligned}
&\mathrm{P}(\theta|\mathrm{Y}) \\
&= \prod_{i=1}^{n} \left[\frac{\pi}{\pi+\rho+\sigma} \mathrm{BetaBinormial}(\mathrm{N},\alpha,\alpha) \right. \\
&+ \frac{\rho}{\pi+\rho+\sigma} \mathrm{BetaBinormial}(\mathrm{N},\beta,\gamma) \\
&\left. + \frac{\sigma}{\pi+\rho+\sigma} Possion(\lambda) \right] \frac{d^c \pi^{c-1} e^{-d\pi}}{\Gamma(c)} \frac{d^c \rho^{c-1} e^{-d\rho}}{\Gamma(c)} \\
&\times \frac{d^c \sigma^{c-1} e^{-d\sigma}}{\Gamma(c)} \frac{f^e \alpha^{e-1} e^{-f\alpha}}{\Gamma(e)} \frac{\beta^{g-1}(1-\beta)^{h-1}}{B(g,h)} \\
&\times \frac{\gamma^{p-1}(1-\gamma)^{q-1}}{B(p,q)} \frac{n^m \lambda^{m-1} e^{-n\lambda}}{\Gamma(m)}
\end{aligned} \tag{13}$$

为了便于计算，我们将等式两边分别取对数，得到如下等式：

$$
\begin{aligned}
\ln P(\theta|Y) = \sum_{i=1}^{n} \ln \Bigg[& \frac{\pi}{\pi+\rho+\sigma} \mathrm{BetaBinormial}(N, \alpha, \alpha) \\
& + \frac{\rho}{\pi+\rho+\sigma} \mathrm{BetaBinormial}(N, \beta, \gamma) \\
& + \frac{\sigma}{\pi+\rho+\sigma} Possion(\lambda) \Bigg] + clnd \\
& + (c-1)ln\pi - d\pi - \Gamma(c) + clnd \\
& + (c-1)ln\rho - d\rho - \Gamma(c) + clnd \\
& + (c-1)ln\sigma - d\sigma - \Gamma(c) + elnf \\
& + (e-1)ln\alpha - f\alpha - \Gamma(c) + (g-1)ln\beta \\
& + (h-1)\ln(1-\beta) - B(g,h) \\
& + (p-1)ln\gamma + (q-1)\ln(1-\gamma) \\
& - B(p,q) + mlnn + (m-1)ln\lambda - n\lambda \\
& - \Gamma(m)
\end{aligned}
\tag{14}
$$

由于七个参数相互独立，因此对于每一个参数，其他参数在分布函数仅作为常数存在，因此，我们可以略去常数，化简后得到每个参数成比例服从于一下分布：

$$
\begin{aligned}
& \ln P(\pi|\rho, \sigma, \alpha, \beta, \gamma, \lambda) \\
& \propto \sum_{i=1}^{n} \ln \Bigg[\frac{\pi}{\pi+\rho+\sigma} \mathrm{BetaBinormial}(N, \alpha, \alpha) \\
& + \frac{\rho}{\pi+\rho+\sigma} \mathrm{BetaBinormial}(N, \beta, \gamma) \\
& + \frac{\sigma}{\pi+\rho+\sigma} Possion(\lambda) \Bigg] + (c-1)ln\pi - d\pi
\end{aligned}
\tag{15}
$$

$$\begin{aligned}&\ln P(\rho|\pi,\sigma,\alpha,\beta,\gamma,\lambda)\\&\propto\sum_{i=1}^{n}\ln\left[\frac{\pi}{\pi+\rho+\sigma}\text{BetaBinormial}(N,\alpha,\alpha)\right.\\&+\frac{\rho}{\pi+\rho+\sigma}\text{BetaBinormial}(N,\beta,\gamma)\\&\left.+\frac{\sigma}{\pi+\rho+\sigma}Possion(\lambda)\right]+(c-1)ln\rho-d\rho\end{aligned}\tag{16}$$

$$\begin{aligned}&\ln P(\sigma|\pi,\rho,\alpha,\beta,\gamma,\lambda)\\&\propto\sum_{i=1}^{n}\ln\left[\frac{\pi}{\pi+\rho+\sigma}\text{BetaBinormial}(N,\alpha,\alpha)\right.\\&+\frac{\rho}{\pi+\rho+\sigma}\text{BetaBinormial}(N,\beta,\gamma)\\&\left.+\frac{\sigma}{\pi+\rho+\sigma}Possion(\lambda)\right]+(c-1)ln\sigma-d\sigma\end{aligned}\tag{17}$$

$$\begin{aligned}&\ln P(\alpha|\pi,\rho,\sigma,\beta,\gamma,\lambda)\\&\propto\sum_{i=1}^{n}\ln\left[\frac{\pi}{\pi+\rho+\sigma}\text{BetaBinormial}(N,\alpha,\alpha)\right.\\&+\frac{\pi}{\pi+\rho+\sigma}\text{BetaBinormial}(N,\beta,\gamma)\\&\left.+\frac{\pi}{\pi+\rho+\sigma}Possion(\lambda)\right]+(e-1)ln\alpha-f\alpha\end{aligned}\tag{18}$$

$$\begin{aligned}&\ln P(\beta|\pi,\rho,\sigma,\alpha,\gamma,\lambda)\\&\propto\sum_{i=1}^{n}\ln\left[\frac{\pi}{\pi+\rho+\sigma}\text{BetaBinormial}(N,\alpha,\alpha)\right.\\&+\frac{\pi}{\pi+\rho+\sigma}\text{BetaBinormial}(N,\beta,\gamma)\\&\left.+\frac{\pi}{\pi+\rho+\sigma}Possion(\lambda)\right]+(g-1)ln\beta\\&+(h-1)\ln(1-\beta)\end{aligned}\tag{19}$$

$$
\begin{aligned}
&\ln P(\gamma|\pi,\rho,\sigma,\alpha,\beta,\lambda) \\
&\propto \sum_{i=1}^{n} \ln\left[\frac{\pi}{\pi+\rho+\sigma}\mathrm{BetaBinormial}(N,\alpha,\alpha)\right. \\
&+\frac{\pi}{\pi+\rho+\sigma}\mathrm{BetaBinormial}(N,\beta,\gamma) \\
&\left.+\frac{\pi}{\pi+\rho+\sigma}Possion(\lambda)\right]+(p-1)ln\gamma \\
&+(q-1)\ln(1-\gamma)
\end{aligned}
\tag{20}
$$

$$
\begin{aligned}
&\ln P(\lambda|\pi,\rho,\sigma,\alpha,\beta,\gamma) \\
&\propto \sum_{i=1}^{n} \ln\left[\frac{\pi}{\pi+\rho+\sigma}\mathrm{BetaBinormial}(N,\alpha,\alpha)\right. \\
&+\frac{\pi}{\pi+\rho+\sigma}\mathrm{BetaBinormial}(N,\beta,\gamma) \\
&\left.+\frac{\pi}{\pi+\rho+\sigma}Possion(\lambda)\right]+(m-1)ln\lambda-n\lambda
\end{aligned}
\tag{21}
$$

随后，我们设定初始值，使用 R 语言编写程序，通过 MCMC 算法[36]模拟出 1000 组有效参数，取有效参数的中位数作为模型的参数值。（附录 4）

我们以 SNP 位点的转录不具有偏好性作为零假设，以 SNP 位点转录具有偏好性作为备择假设。在获得参数值后，我们将观测值带入已经求得参数的分布函数中，使用 R 语言编程通过以下公式计算 p 值（附录 5）：

$$
\begin{aligned}
p=&\frac{\pi}{\pi+\rho+\sigma}\mathrm{BetaBinormial}(N,\alpha,\alpha) \\
&/\left[\frac{\pi}{\pi+\rho+\sigma}\mathrm{BetaBinormial}(N,\alpha,\alpha)\right. \\
&+\frac{\rho}{\pi+\rho+\sigma}\mathrm{BetaBinormial}(N,\beta,\gamma) \\
&\left.+\frac{\sigma}{\pi+\rho+\sigma}Possion(\lambda)\right]
\end{aligned}
\tag{22}
$$

随后，我们对 p 值进行校正。如果校正后的 p 值小于 0.05，则我们认为在该位点处拒绝零假设，即该 SNP 位点具有转录偏好性，即该 SNP 位点所对应的

m^6A 位点具有转录偏好性。

3.2.5 模型检验

为了检验我们设计的模型是否具有可靠性，我们使用 R 语言模拟出具有偏好性以及不具有偏好性的模拟数据集，将模拟数据集中的数据带入模型中，仿照我们对真实数据集的处理方式对模型进行检验。

我们首先使用均匀分布模拟出来自于父本及母本每个位点总的转录数，其实从我们将其作为 N 值以 0.5 及 0.2 作为 p 值分别模拟出服从二项分布的具有偏好性的样本以及服从二项分布的不具有偏好性的样本，作为模拟数据集。我们将具有偏好性的样本比例分别设置为 10%、20%、30%、40%、50%、60%、70%、80%对模型进行多次检验，以检验模型的有效性及在不同情况下的性能。（附录 6）

4. 结果

通过以上描述的实验过程，我们获得各个实验的结果，包括 m^6A 位点的峰值区域、预测出的 m^6A 位点位点、预测出的 SNP 位点以及经检验具有等位基因特异性的 m^6A 位点。

4.1 m^6A 峰值鉴定

使用 MeTPeak 包[20]对测序数据进行分析后，总共获得 51783 例数据。使用 MACS[26]软件对测序数据进行分析后，总共获得 3649 例数据。使用 MEME 软件对 m^6A 位点周围的碱基情况进行描绘，得到图 3。其中，核苷酸的高度代表其出现的频率。

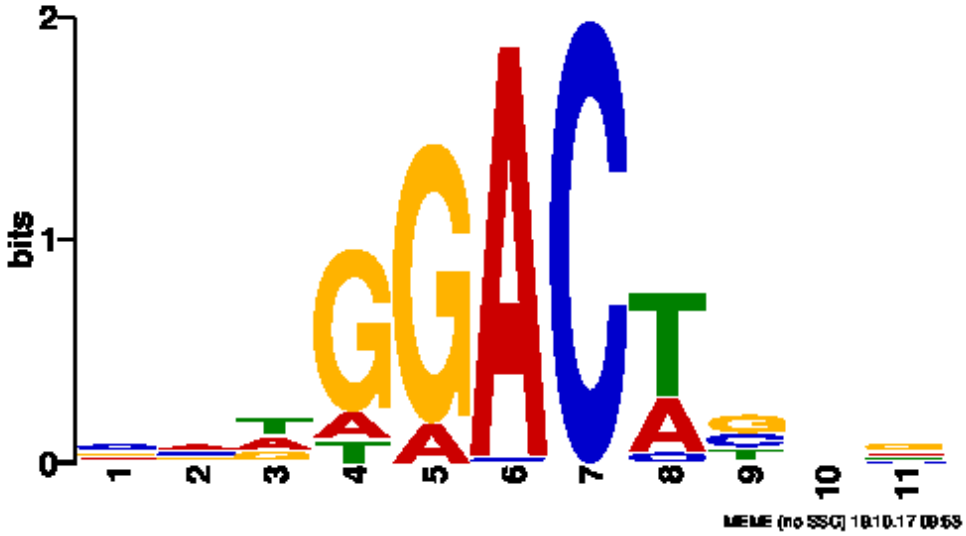

图 3 m^{6}A 位点峰值区域碱基分布。核苷酸的高度代表其出现的频率

Figure 3 Distribution of nucleotides in peak regions. The height of nucleotides represents their frequency

由图可知，m^{6}A 位点附近的核苷酸序列具有一定模式。在修饰位点及其周围的碱基序列是“GGACT”的可能性较高。

4.2 m^{6}A 位点预测

对于第一组数据，我们使用 SRAMP 对获得的 m^{6}A 峰值区域进行分析，获得不同置信度的 m^{6}A 单碱基精度的预测位点。其中，评分位于 0.52-0.557 之间的 m^{6}A 位点被判定为具有低置信度的 m^{6}A 位点，评分位于 0.557-0.60 之间的 m^{6}A 位点被判定为具有中等置信度的 m^{6}A 位点，评分位于 0.60-0.672 之间的 m^{6}A 位点被判定为具有高置信度的 m^{6}A 位点，评分位于 0.672 以上的 m^{6}A 位点被判定为具有非常高置信度的 m^{6}A 位点。对使用 MACS 获得 m^{6}A 峰值区域进行位点预测后，获得的不同置信度的 m^{6}A 位点数目如下表所示：

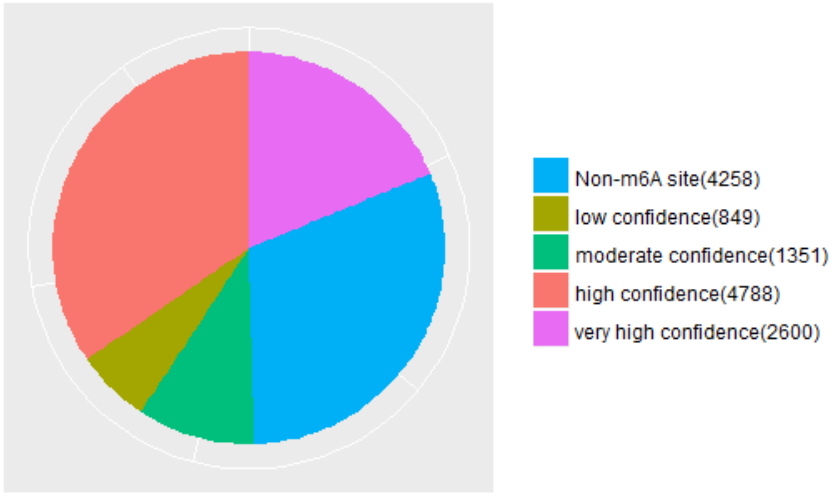

图 4 第一组数据使用 MACS 获得峰值区域的 m6A 位点预测结果

Figure 4 The results of SRAMP on the region given by MACS in first data set

对于使用 MeTPeaks 包获得的 m^6A 峰值区域，我们同样使用 SRAMP 进行位点预测，得到不同置信度的 m^6A 位点数目如下表所示：

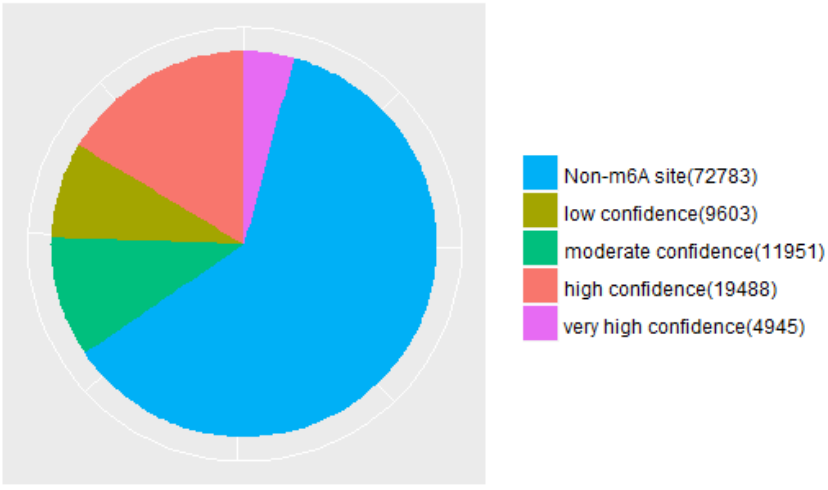

图 5 第一组数据使用 MeTPeak 获得峰值区域的 m^6A 位点预测结果

Figure 5 The results of SRAMP on the region given by MeTPeak in first data set

4.3 SNP 位点

首先，我们对预测出的 SNP 位点进行筛选，以 20 为阈值筛选出错误率小于 1%的 SNP 位点。对于第一组数据，我们共获得 445571 个 SNP 位点。对于第二组数据，我们共获得 14776 个 SNP 位点。

然后，我们使用 ANNOVAR[37]对预测出来的 SNP 位点进行注释，获得 SNP 位点信息如下：

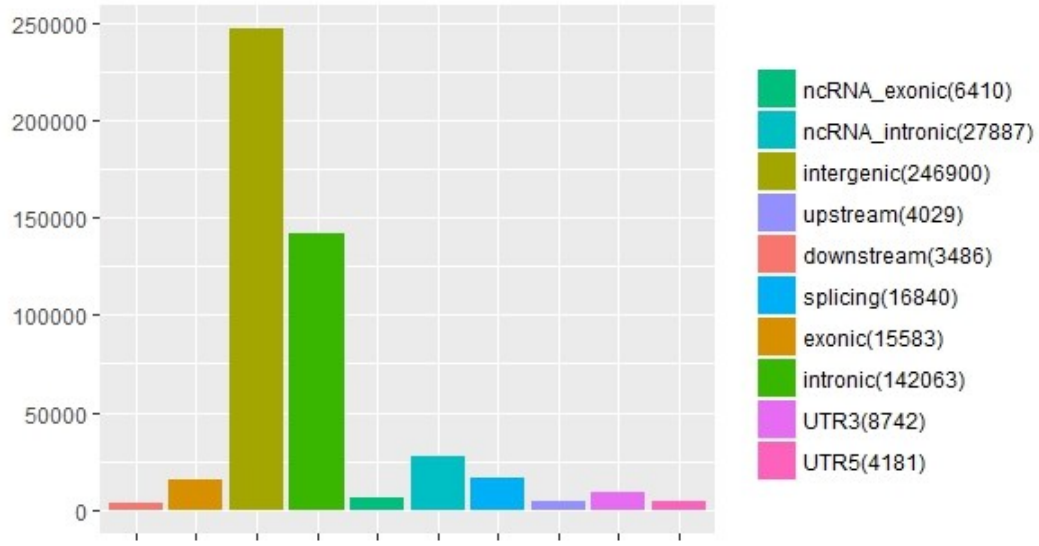

图 6 第一组数据中 SNP 位点位于个功能区域数目

Figure 6 Numbers of SNP in different function area of first data set

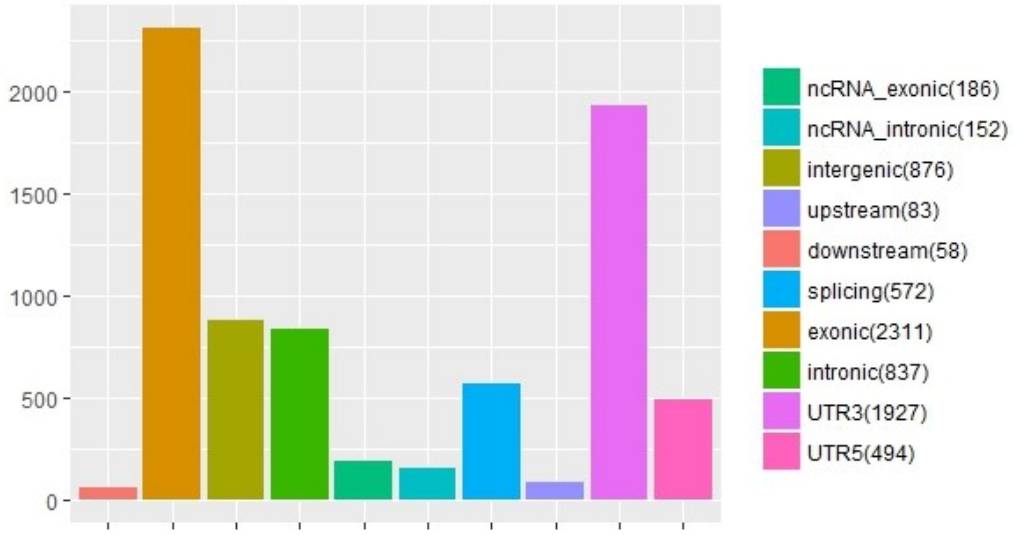

图 7 第二组数据中 SNP 位点位于个功能区域数目

Figure 7 Numbers of SNP in different function area of second data set

4.4 测定 m^{6}A 位点是否具有等位基因特异性

我们使用 R 语言构建模型，随后使用模拟数据集对模型进行检验，以判断我们构建出来的模型是否能够有效地检验出具有等位基因特异性的 m^{6}A 位点。若模型对于位点是否具有转录偏好性的检验效果较出色，则我们利用该模型对我们的数据进行分析。

4.4.1 模型检验结果

我们使用模拟数据集对模型进行检验，通过比较模型对于每一个位点模拟数据的判定与其真实情况的差距来判断模型是否具有准确性。仅有一个作为具有偏好性的位点被模型判定为不具有偏好性。

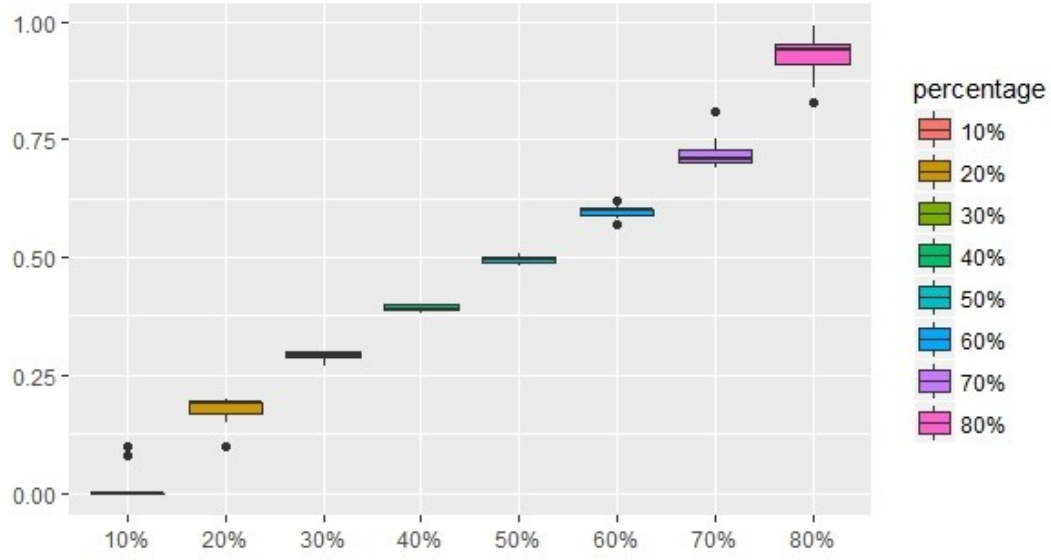

图 8 模型测试箱图。

其中 x 轴代表模拟数据集中具有特异性的位点百分比，y 轴代表测试结果中具有特异性的位点比例。

Figure 8 The box plot shows results of model test.

Numbers in x-axis shows percentage of the significant data, numbers in y-axis represent rate of significant data in result.

由箱图我们可以看出，该模型在特异性位点的比例为 30%到 70%时，性能稳定且准确率高，但是若特异性位点的比例过低或过高，则判断率及准确率均会下降。

4.4.2 m^6A 位点检验结果

我们使用该模型对 IP 样本数据和 Input 样本数据进行了检验。对于每一个样本，我们先使用它的 total reads 和 reads count 作为 N 值和观测值 yi，利用 MCMC 计算出其服从的分布的参数，再利用获得的参数计算 p 值，将 p 值小于 0.05 的位点作为具有等位基因特异性的位点。

对于第一组及第二组样本数据，我们所获得 SNP 位点、m^6A 位点、观测值以及具有等位基因特异性的 m^6A 位点数的结果如下表所示：

表格 1 实验数据统计结果

Table 1 Indexes of experiment

	SNP 位点数	m^6A 位点数	观测值数	具有等位基因的 m^6A 位点数
第一组数据	443087	31439	3020	5
第二组数据	7234	33653	997	6

对于第一个真实数据集，我们总共获得 3020 个位点数据，其中 6 个位点具有等位基因特异性。对于第二个真实数据集，总共获得 3020 个位点数据，其中 6 个位点具有等位基因特异性。具有等位基因特异性的位点信息如下表所示，其中 chrom 代表 m^6A 以及 SNP 位点在哪条染色体上，m^6A 代表 m^6A 位点在该条染色体上的位置，SNP 代表 SNP 位点在该条染色体上的位置，IP_p 代表根据模型计算出的 p 值，whesig 代表该位点是否具有等位基因特异性：

表格 2 第一组数据中具有等位基因特异性的 m^6A

Table 2 Allele specific m^6A sites of first data set

chrom	m^6A	SNP	IP_p	whesig	refGene
chr5	179048325	179048331	0.043595	significant	HNRNPH1
chr2	239007263	239007278	0.048595	significant	UBE2F-SCLY
chr2	239007277	239007278	0.048595	significant	UBE2F-SCLY

表格 3 第二组数据中具有等位基因特异性的 m^6A

Table 3 Allele specific m^6A sites of second data set

chrom	m^6A	SNP	IP_p	whesig	refGene
chr5	118729016	118729013	0.037736	significant	TNFAIP8
chr5	118729031	118729013	0.037736	significant	TNFAIP8
chr1	206906136	206906119	0.042394	significant	MAPKAPK2
chr1	206906141	206906119	0.042394	significant	MAPKAPK2
chr3	145788479	145788467	0.04282	significant	PLOD2
chr3	14183802	14183792	0.049151	significant	TMEM43

5. 讨论

在本次实验中，我们总共使用了两组来源不同的测序数据进行分析，每组数据均包括三个 IP 样本和三个 Input 样本。对于每一组测序数据，我们使用了 MACS 以及 MeTPeak 包两种方法进行 m^6A 位点的峰值预测。两组不同来源的测序数据的使用帮助我们进一步认识具有等位基因特异性的 m^6A 位点出现的规律，而使用两种方法进行峰值预测也加深我们对 m^6A 位点峰值预测方法的了解。

5.1 m^6A 位点特异性结果讨论

我们根据 PNA 转录特性构建出 reads count 服从的分布，并使用贝叶斯方法和 MCMC 算法得到模型的参数值。然后根据分布，利用统计学方法判断每一个 SNP 位点对应的 m^6A 位点是否具有等位基因特异性。

为了检测这一方法的可靠性，我们首先构建出具有偏好性的样本比例为 10%、20%、30%、40%、50%、60%、70%、80%的模拟数据集对模型进行多次检验。结果显示，模型在具有偏好性的样本比例过高或者过低的情况下，对位点的检测率不稳定，而在其他情况下具有良好的准确率及稳定性。

在检验了模型的性能后，我们使用该模型对我们的获得的真实数据进行检测，结果显示，两组数据均检测到具有等位基因特异性的 m^6A 位点，分别为 3 个与 6 个。从结果上看，由于具有偏好性位点比例较低，因此，所检验出的具有等位基因特异的 m^6A 位点数目可能低于实际值。未来我们可以针对具有偏好性的样本比例过高或者过低的情况，对模型进行进一步改良。

同时，考虑到两组数据的观测值数量差距较大，分别为 3020 个观测值以及 997 个观测值，得到的具有等位基因特异性的 m^6A 位点数目却差距不大，有可能观测值的多少与具有等位基因特异性的 m^6A 位点数目不成正相关关系。两组测序数据中比例差异的来源可能是由于细胞的不同，也可能是由于样品的原因。由于对于 m^6A 的研究十分有限，而对于 m^6A 位点偏好性的研究尚未报道，因此我们无法对这一现象做进一步分析解释。

5.2 两种 m^{6}A 峰值区域预测结果比较

在获得 m^{6}A 位点经过免疫沉淀法后获得的高通量测序数据后，我们使用了两款软件对测序数据进行了 m^{6}A 峰值区域鉴定以及位点预测，结果有差异较大。

首先，对于测定 m^{6}A 的峰值区域，使用 MeTPeak 包[20]对测序数据进行分析后，总共获得 51783 例数据。使用 MACS[26]软件对测序数据进行分析后，总共获得 3649 例数据。使用 MeTPeak 包大大增加了 m^{6}A 峰值区域的检测率。考虑到 MeTPeak 使用了更为父子的模型，引入更多的参数进行预测，它对于 m^{6}A 峰值区域的敏感性可能更高，从而对于峰值较为不显著的峰值区域也能预测出来。同时，MACS 是通过先找到峰值最高点，再以其为中心划定峰值区域的，所以如果 IP 和 Input 样本在峰处不明显，也会导致峰值区域鉴定的不敏感性。

此外，我们观察使用两种方式获得的峰值区域中 m^{6}A 位点的预测结果可以发现，使用 MACS 软件预测出了 3649 个峰值区域，在其中预测出包括 849 个具有低置信度的 m^{6}A 位点，占 6.13%，1351 个具有中等置信度的 m^{6}A 位点，占 9.76%，4788 个具有高置信度的 m^{6}A 位点，占 34.58%，以及 2600 个具有极高置信度的 m^{6}A 位点，占 18.78%。而使用 MeTPeak 包预测出了 51783 个 m^{6}A 峰值区域，在内预测出包括 9603 个具有低置信度的 m^{6}A 位点，占 8.09%，11951 个具有中等置信度的 m^{6}A 位点，占 10.06%，19488 个具有高置信度的 m^{6}A 位点，占 16.41%，以及 4945 个具有极高置信度的 m^{6}A 位点，占 4.16%。

我们可以看出，对于两种方法预测出的峰值区域，在其中预测出的具有中等置信度的 m^{6}A 位点以及具有高置信度的 m^{6}A 位点的比例较为接近，而被鉴定为非 m^{6}A 位点的比例则差异较大，在由 MACS 检测出的峰值区域内，仅有 30.75%为非 m^{6}A 位点，而在由 MeTPeaks 检测出的峰值区域内，有 61.28%的位点被检测为非 m^{6}A 位点。因此对于后者，具有高置信度及极高置信度的 m^{6}A 位点的比例降低。

因为 MeTPeaks 对于 m^{6}A 位点峰值区域更为敏感，因此它鉴定出的峰值区域远大于 MACS 鉴定出的峰值区域，因此该区域内会有更多的非 m^{6}A 位点。而两者预测出相似比例的具有中等置信度以及低置信度的 m^{6}A 位点说明两者对于这两类 m^{6}A 位点的敏感程度相似，同时中等置信度以及低置信度的 m^{6}A 位点的数量与峰值区域的大小成正相关关系。而具有高置信度以及极高置信度的 m^{6}A 位点则不同，

它们的数量相对有限，不容易受峰值区域范围的影响。因此，随着峰值区域的增加，预测出的 m^{6}A 位点数目的增加频率大大减少。这也侧面反映了使用 MACS 对通过免疫沉淀法获得的高通量测序数据的 m^{6}A 位点峰值区域的分析检测是较为准确的。

但是，如果我们直接观察 m^{6}A 位点的预测数目能够发现，尽管高置信度以及极高置信度的 m^{6}A 位点的比例较低，使用 MeTPeaks 获得的峰值区域中预测出的这两类 m^{6}A 位点的数目仍远大于使用 MACS 预测出的峰值区域内这两种 m^{6}A 位点的数目。由此可见，使用 MACS 对 m^{6}A 位点的峰值区域进行预测这一方法的敏感性较为缺乏。

由于本实验旨在利用统计学知识，构建模型对 m^{6}A 位点是否具有等位基因特异性进行判断，因此我们需要大量的数据来获得统计学规律，因此我们使用了通过 MeTPeaks 或的 m^{6}A 位点峰值区域内的 m^{6}A 预测位点进行后续实验。

参考文献：

[1] Batista P J, Molinie B, Wang J, et al. m(6)A RNA modification controls cell fate transition in mammalian embryonic stem cells[J]. Cell Stem Cell,2014,15(6):707.

[2] Zhong S, Li H, Bodi Z, et al. MTA Is an Arabidopsis Messenger RNA Adenosine Methylase and Interacts with a Homolog of a Sex-Specific Splicing Factor[J]. Plant Cell,2008,20(5):1278.

[3] Ping X L, Sun B F, Wang L, et al. Mammalian WTAP is a regulatory subunit of the RNA N6-methyladenosine methyltransferase[J]. Cell Research,2014,24(2):177.

[4] Wang D G, Fan J B, Siao C J, et al. Large-Scale Identification, Mapping, and Genotyping of Single-Nucleotide Polymorphisms in the Human Genome[C]. 1998.

[5] Sekar K, Voight B F, Shaun P, et al. Genome-wide association of early-onset myocardial infarction with common single nucleotide polymorphisms, common copy number variants, and rare copy number variants[J]. Nature Genetics,2009,41(3):334.

[6] He C, Chen L, Simmons M, et al. Putative SNP discovery in interspecific hybrids of catfish by comparative EST analysis[J]. Animal Genetics,2003,34(6):445-448.

[7] Glenn K L, Grapes L, Suwanasopee T, et al. SNP analysis of AMY2 and CTSL genes in Litopenaeus

vannamei and Penaeus monodon shrimp.[J]. Animal Genetics,2015,36(3):235-236.

[8] Pastinen T. Genome-wide allele-specific analysis: insights into regulatory variation.[J]. Nature Reviews Genetics,2010,11(8):533-538.

[9] Sigurdsson M I, Saddic L, Heydarpour M, et al. Allele-specific expression in the human heart and its application to postoperative atrial fibrillation and myocardial ischemia[J]. Genome Medicine,2016,8(1):127.

[10] Qin S, Li Q, Zhou J, et al. Homeostatic maintenance of allele-specific p16 methylation in cancer cells accompanied by dynamic focal methylation and hydroxymethylation.[J]. Plos One,2014,9(5):e97785.

[11] Pastinen T. Genome-wide allele-specific analysis: insights into regulatory variation.[J]. Nature Reviews Genetics,2010,11(8):533-538.

[12] Hellman A, Chess A. Gene body-specific methylation on the active X chromosome.[J]. Science,2007,315(5815):1141.

[13] Zhang Y, Rohde C, Reinhardt R, et al. Non-imprinted allele-specific DNA methylation on human autosomes[J]. Genome Biology,2009,10(12):R138.

[14] Audic S, Claverie J M. The significance of digital gene expression profiles.[J]. Genome Research,1997,7(10):986-995.

[15] Meng J, Cui X, Rao M K, et al. Exome-based analysis for RNA epigenome sequencing data.[J]. Bioinformatics,2013,29(12):1565-1567.

[16] Meng J, Lu Z, Liu H, et al. A protocol for RNA methylation differential analysis with MeRIP-Seq data and exomePeak R/Bioconductor package.[J]. Methods,2014,69(3):274-281.

[17] Mayba O, Gilbert H N, Liu J, et al. MBASED: allele-specific expression detection in cancer tissues and cell lines[J]. Genome Biology,2014,15(8):405.

[18] Emerson J J, Hsieh L C, Sung H M, et al. Natural selection on cis and trans regulation in yeasts[J]. Genome Research,2010,20(6):826-836.

[19] Dominissini D, Moshitchmoshkovitz S, Salmondivon M, et al. Transcriptome-wide mapping of N(6)-methyladenosine by m(6)A-seq based on immunocapturing and massively parallel sequencing.[J]. Nature Protocols,2013,8(1):176-189.

[20] Cui X, Jia M, Zhang S, et al. A novel algorithm for calling mRNA m6A peaks by modeling

biological variances in MeRIP-seq data[J]. Bioinformatics,2016,32(12):i378.

[21] Zhou Y, Zeng P, Li Y H, et al. SRAMP: prediction of mammalian N6-methyladenosine (m6A) sites based on sequence-derived features[J]. Nucleic Acids Research,2016,44(10):e91.

[22] Garrison E, Marth G. Haplotype-based variant detection from short-read sequencing[J]. Quantitative Biology,2012.

[23] Ke S, Alemu E A, Mertens C, et al. A majority of m6A residues are in the last exons, allowing the potential for 3' UTR regulation[J]. Genes & Development,2015,29(19):2037-2053.

[24] SRA ToolKit[Z].

[25] Dominissini D, Moshitch-Moshkovitz S, Salmon-Divon M, et al. Transcriptome-wide mapping of N(6)-methyladenosine by m(6)A-seq based on immunocapturing and massively parallel sequencing.[J]. Nature Protocols,2013,8(1):176-189.

[26] Yong Z, Tao L, Meyer C A, et al. Model-based Analysis of ChIP-Seq (MACS)[J]. Genome Biology,2008,9(9):1-9.

[27] 李语丽，于军，宋述慧. RNA中6-甲基腺嘌呤的研究进展[J]. 遗传,2013,35(12):1340-1351.

[28] MACS[Z]. http://liulab.dfci.harvard.edu/MACS/

[29] bowtie[Z]. http://bioconda.github.io/recipes/bowtie/README.html

[30] bedtools[Z]. https://github.com/arq5x/bedtools2

[31] Quinlan A R, Hall I M. BEDTools: a flexible suite of utilities for comparing genomic features.[J]. Bioinformatics,2010,26(6):841.

[32] SAMtools[Z]. http://samtools.sourceforge.net/

[33] MeTPeak[Z]. https://github.com/compgenomics/MeTPeak

[34] SRAMP[Z]. http://www.cuilab.cn/sramp/

[35] freebayes[Z]. https://github.com/ekg/freebayes

[36] Andrieu C, Freitas N D, Doucet A, et al. An Introduction to MCMC for Machine Learning[J]. Machine Learning,2003,50(1-2):5-43.

[37] Wang K, Li M, Hakonarson H. ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data[J]. Nucleic Acids Research,2010,38(16):e164.

致谢：

感谢任间老师对于本毕业设计的精心及细致指导。任老师非常关心本科生在实验室的学习。在进入实验室时，任老师就耐心地询问了我未来的计划以及想要发展的方向，提出宝贵的意见。在进入实验室的期间，老师也多次询问其我的毕业设计的完成情况，提供建议并鼓励我多与师兄师姐交流。感谢任老师，您的支持与鼓励促使我一直努力并最终完成毕业设计。

感谢谢宇斌及实验室的其他师兄师姐在本毕业设计完成过程中提供的技术及相关知识的帮助。在我进行实验时，曾多次遇到 bug 无法解决，是实验室的师兄师姐不厌其烦地帮助我进行调试，寻找解决 bug 的方法。在我的实验进入瓶颈期，无法进行时，是谢宇斌师兄耐心地指点我，给我提供思路及方法。没有实验室师兄及师姐的帮助，我无法如此顺利地完成我的毕业设计。

最后，我还要感谢我的辅导员王莹、我的舍友以及我的父母。感谢她们在我完成毕业设计的过程中，在我遇到困难或心情低落的时候，倾听我、鼓励我、指导我、陪伴我。感谢她们一直给予我的爱与包容。

附录：

附录1 获得 m6A 位点所需信息

```
# -*- coding: utf-8 -*


#step_4
#在 sramp 的结果文件（已过滤出有效部分）中得到 m6A 位点
#0.53 获得有置信度，0.557 有中等置信度


input_file='MeTPeaks_sramp_pos_2.txt'      #输入文件
output_file='MeTPeaks_m6A_posi.txt'    #输出文件


import os
import re
os.getcwd()     #获取当前工作目录
os.chdir("C:\Users\Administrator\Desktop\m6A\Python file")     #修改当
前工作目录


lineL=[]     #用于储存 sramp 文件的信息
#i=0
for line in open(input_file,"r"):
    if not line.startswith('#'):
        #tepL1 = []  # 储存未处理的 SNP 信息
        #tepL2 = []  # 储存分隔的 INFO
        #tepL3 = []  # 处理好的 SNP 信息
        #print tepL2 = []  # 打印
        #print tepL3 = []  # 打印
        posL=[]  #用于存储每一个位点的信息
        tepL=[]  #临时列表
        posL=line.split()
        tepL=re.split('[:-]',posL[0],2)
        tepL.append(posL[1])
        #print i
        #i = i + 1
        print tepL[0]
        a=int(tepL[1])
```

```
        b=int(tepL[3])
        print a
        print b
        c=a+b
        tepL.append(str(c))
        lineL.append(tepL)
        #m = int(tepL[1])
        #print lineL[i]

print 'lineL', len(lineL)
for i in range(0,6):
    print lineL[i]

fileObject = open(output_file, 'w')
fileObject.write('#chrom position')
fileObject.write('\n')
for ip in lineL:
    fileObject.write(ip[0])
    fileObject.write('\t')
    fileObject.write(ip[4])
    fileObject.write('\n')
fileObject.close()
```

附录2 获得 SNP 位点所需信息

```
# -*- coding: utf-8 -*


import os
import re

os.getcwd()     #获取当前工作目录
os.chdir("C:\Users\Administrator\Desktop\m6A\Python file")     #修改当
前工作目录

input_file='Input_pos_new.vcf'      #输入文件
output_file='Input_SNP_info_new.txt'     #输出文件

#IP 样本
#获得 SNP 位点
SNP_L=[]    #用于储存 SNP 的位置
for line in open(input_file):
```

```
    if not line.startswith('#'):
        tepL1 = []  # 储存未处理的 SNP 信息
        tepL2 = []  # 储存分隔的 INFO
        tepL3 = []  # 处理好的 SNP 信息
        tepL1=line.split()
        tepL2=tepL1[4].split(';')
        DPL = tepL2[7].split('=')
        ROL = tepL2[28].split('=')
        AOL = re.split('[=,]', tepL2[5])
        #ROL = re.split('[=,]', tepL2[28])
        #print tepL2
        tepL3.append(tepL1[0])  # 提取出 chrom, pos, ref, alt
        tepL3.append(tepL1[1])
        tepL3.append(tepL1[2])
        tepL3.append(tepL1[3])
        tepL3.append(AOL[1])
        tepL3.append(DPL[1])
        tepL3.append(ROL[1])
        if len(AOL)!=2:
            #print tepL2
            if  int(AOL[1])>int(AOL[2]):
                tepL3.append(AOL[1])
            else:
                tepL3.append(AOL[2])
        else:
            tepL3.append(AOL[1])
        SNP_L.append(tepL3)
for i in range(0,6):
    print SNP_L[i]

#写入文件
fileObject = open(output_file, 'w')
fileObject.write('#chrom  position  ref  alt  AO  DP  RO  yi')
fileObject.write('\n')
for ip in SNP_L:
    for i in ip:
        fileObject.write(i)
        fileObject.write('\t')
    '''fileObject.write(ip[1])
    fileObject.write('\t')
    fileObject.write(ip[2])
    fileObject.write('\t')
    fileObject.write(ip[3])
```

```
    fileObject.write('\t')
    fileObject.write(ip[4])'''
    fileObject.write('\n')
fileObject.close()
```

附录3 将 m6A 位点与 SNP 位点匹配

```
#-*- coding:utf-8 -*-

#毕设 m6A
#使用的均为有效数据，m6A 的阈值为 0.557，SNP 的阈值为 20（错误率 1%）
#第四步：判定等为基因特异性
#1、提取出位于 m6A 位点周围的 SNP 位点
'''
重要的数据：
1、m6A_L：用于存储 m6A 位点信息；
2、SNP_L：用于存储 SNP 位点信息；
3、SNP_aro_L：用于存储位于 m6A 位点周围的 SNP 位点。'''

m6A_file='MeTPeaks_m6A_posi.txt'
SNP_file='IP_SNP_info.txt'


import os

#设置路径
os.getcwd()
os.chdir("C:\Users\Administrator\Desktop\m6A\Python file")

#Input 样本
#获得 m6A 位点
m6A_L=[]
for line in open(m6A_file):
    if not line.startswith('#'):
        m6A_L.append(line.split())

for i in range(0, 6):
    print m6A_L[i]
print 'm6A_L:',len(m6A_L)

#获得 SNP 位点
SNP_L=[]    #用于储存 SNP 的位置
```

```
for line in open(SNP_file):
    if not line.startswith('#'):
        SNP_L.append(line.split())
for i in range(0,6):
    print SNP_L[i]
print "SNP_L:",len(SNP_L)

#以 m6A 位点作为锚点，找到周围的 SNP 位点
SNP_aro_L=[]    #用于存储 m6A 位点周围的 SNP 位点
numL=[]         #储存 N,yi
numN_L=[]         #储存 N
numY_L=[]         #储存 yi
LogL=[]   #结果列表
tepL=[]      #临时储存
for m in m6A_L:
#for i in range(0,1000):    #以前 1000 个作为测试样本
    #m=m6A_L[i]
    for s in SNP_L:
        if m[0]==s[0]:
            if int(s[1]) in range(int(m[1])-25,int(m[1])+25):
                tepL1=[]
                tepL2=[]            #储存 N，yi 信息
                tepL1.append(m[0])
                tepL1.append(m[1])
                tepL1.append(s[1])
                tepL1.append(s[2])
                tepL1.append(s[3])
                tepL1.append(s[5])
                tepL1.append(s[7])
                SNP_aro_L.append(tepL1)
                tepL2.append(s[5])
                tepL2.append(s[7])
                numL.append(tepL2)
                numN_L.append(s[5])
                numY_L.append(s[7])


for i in range(0,6):
    print SNP_aro_L[i]
print "SNP_aro_L:",len(SNP_aro_L)

for i in range(0,6):
    print numY_L[i]
```

```
print "numY_L:",len(numY_L)

for i in range(0,6):
    print numN_L[i]
print "numN_L:",len(numN_L)

#把 SNP_aro_L 写入文件
fileObject = open('IP_SNP_aro_L_25.txt', 'w')
fileObject.write('#chr m6A SNP ref alt N yi')
fileObject.write('\n')
for ip in SNP_aro_L:
    for i in ip:
        fileObject.write(str(i))
        fileObject.write('\t')
    fileObject.write('\n')
fileObject.close()

#把 numL 写入文件
fileObject = open('IP_numL_25.txt', 'w')
fileObject.write('#N yi')
fileObject.write('\n')
for ip in numL:
    for i in ip:
        fileObject.write(str(i))
        fileObject.write('\t')
    fileObject.write('\n')
fileObject.close()

#把 numN 写入文件
fileObject = open('IP_numN_25.txt', 'w')
for ip in numN_L:
    fileObject.write(ip)
    fileObject.write('\t')
fileObject.close()

#把 numN 写入文件
fileObject = open('IP_numY_25.txt', 'w')
for ip in numY_L:
    fileObject.write(ip)
    fileObject.write('\t')
fileObject.close()
```

附录4 使用 MCMCM 算法求模型参数值

```
library(TailRank)

Input_sample   <-   read.table("Input_SNP_aro_L_25.txt",   sep="\t",
header=F)   #get data
names(Input_sample)<-c('chrom','m6A','SNP','ref','alt','N','yi')
#Input_sample <- Input_sample[1:100,]     #for the test

#y1 <- runif(200,0,200)     #先做一组随机数作为实验
#Y <- floor(y1)

##using MCMC to estimate seven parameters, using normal distribution
to choose sample
##write the common part in advance
Pthetay<-function(yi,ni){      #here need to define all parameters in
advance
  partP<-
pai/(pai+rho+sig)*pbb(yi,ni,alp,alp)+rho/(pai+rho+sig)*pbb(yi,ni,beta
,gam)+sig/(pai+rho+sig)*ppois(yi,lam)
  return(log(partP))
}
#model1<-function(NUMIT=10000,Y=Input_sample[,7],N=Input_sample[,6]){
model1<-function(NUMIT=10000,Y,N){
  ##already given parameters
  c <- 9
  d <- 2
  e <- 9
  f <- 2
  g <- 8
  h <- 1.5
  p <- 8
  q <- 1.5
  m <- 0.5
  n <- 1
  ##create matrix to contain MCMC results
  mchain<-matrix(NA,nr=7,nc=NUMIT)
  #mchain[,1]<-c(1,1,1,0.5,0.5,0.5,0.1)     #set the initial data
  mchain[,1]<-c(0.5,0.5,0.5,5,0.2,0.1,0.1)     #set the initial data
  for(i in 2:NUMIT)
  {
    curpai<-mchain[1,i-1]   #用英文代表希腊字母
    currho<-mchain[2,i-1]
```

```
    cursig<-mchain[3,i-1]
    curalp<-mchain[4,i-1]
    curbeta<-mchain[5,i-1]
    curgam<-mchain[6,i-1]
    curlam<-mchain[7,i-1]
    ##using MH to get significant sample
    propai<-abs(curpai+rnorm(1,0,1)) #draw one sample at random from
N(0,1)
    prorho<-abs(currho+rnorm(1,0,1))
    prosig<-abs(cursig+rnorm(1,0,1))
    proalp<-abs(curalp+rnorm(1,0,1))
    # probeta<-abs(curbeta+rnorm(1,0,1))
    # progam<-abs(curgam+rnorm(1,0,1))
    probeta<-runif(1,0,1)
    progam<-runif(1,0,1)
    prolam<-abs(curlam+rnorm(1,0,1))
    ##Matropolis accept-reject step(in log scale)
    ##initial state
    pai<-curpai
    rho<-currho
    sig<-cursig
    alp<-curalp
    beta<-curbeta
    gam<-curgam
    lam<-curlam
    comb_initial <- sum(as.numeric(mapply(Pthetay,Y,N)))
    ##accept/reject step
    ##pai part
    initial_logpai<-comb_initial+(c-1)*log(pai)-d*pai          #P   of
initial pai
    pai<-propai
    comb <- sum(as.numeric(mapply(Pthetay,Y,N)))
    next_logpai<-comb+(c-1)*log(pai)-d*pai       #P of possible pai
    logpai<-min(0,next_logpai-initial_logpai)
    if (log(runif(1))>logpai)
    {
      pai<-curpai
    }
    ##rho part
    initial_logrho<-comb_initial+(c-1)*log(rho)-d*rho          #P   of
initial rho
    rho<-prorho
    comb <- sum(as.numeric(mapply(Pthetay,Y,N)))
```

```
    next_logrho<-comb+(c-1)*log(rho)-d*rho       #P of possible rho
    logrho<-min(0,next_logrho-initial_logrho)
    if (log(runif(1))>logrho)
    {
      rho<-currho
    }
    ##sig part
    initial_logsig<-comb_initial+(c-1)*log(sig)-d*sig          #P   of
initial sig
    sig<-prosig
    comb <- sum(as.numeric(mapply(Pthetay,Y,N)))
    next_logsig<-comb+(c-1)*log(sig)-d*sig       #P of possible sig
    logsig<-min(0,next_logsig-initial_logsig)
    if (log(runif(1))>logsig)
    {
      sig<-cursig
    }
    ##alp part
    initial_logalp<-comb_initial+(e-1)*log(alp)-f*alp          #P   of
initial alp
    alp<-proalp
    comb <- sum(as.numeric(mapply(Pthetay,Y,N)))
    next_logalp<-comb+(e-1)*log(alp)-f*alp       #P of possible alp
    logalp<-min(0,next_logalp-initial_logalp)
    if (log(runif(1))>logalp)
    {
      alp<-curalp
    }
    ##beta part
    #if(probeta>0&probeta<1){
      initial_logbeta<-comb_initial+(g-1)*log(beta)+(h-1)*log(1-beta)
#P of initial beta
      beta<-probeta
      comb <- sum(as.numeric(mapply(Pthetay,Y,N)))
      next_logbeta<-comb+(g-1)*log(beta)+(h-1)*log(1-beta)         #P
of possible beta
      logbeta<-min(0,next_logbeta-initial_logbeta)
      if (log(runif(1))>logbeta)
      {
        beta<-curbeta
        }
    #}
    ##gam part
```

```
    #if(progam>0&progam<1){
      initial_loggam<-comb_initial+(p-1)*log(gam)+(q-1)*log(1-gam)
#P of initial gam
      gam<-progam
      comb <- sum(as.numeric(mapply(Pthetay,Y,N)))
      next_loggam<-comb+(p-1)*log(gam)+(q-1)*log(1-gam)          #P of
possible gam
      loggam<-min(0,next_loggam-initial_loggam)
      if (log(runif(1))>loggam){
        gam<-curgam
        }
    #}
    ##lam part
    initial_loglam<-comb_initial+(m-1)*log(lam)-n*lam          #P   of
initial lam
    lam<-prolam
    comb <- sum(as.numeric(mapply(Pthetay,Y,N)))
    next_loglam<-comb+(m-1)*log(lam)-n*lam       #P of possible lam
    loglam<-min(0,next_loglam-initial_loglam)
    if (log(runif(1))>loglam)
    {
      lam<-curlam
    }
    ## update chain with new values
    mchain[,i]=c(pai,rho,sig,alp,beta,gam,lam)
  }
  mchain<-mchain[,(ncol(mchain)-99):ncol(mchain)]       #abandon fist
100 groups of parameters
  return(mchain)
}

results<-apply(model1(Y=Input_sample[,7],N=Input_sample[,6]),1,median)
#取列中位数
```

附录5 使用分布函数求 p 值

```
library(TailRank)

Input_sample   <-   read.table("Input_SNP_aro_L_25.txt",   sep="\t",
header=F)   #get data
names(Input_sample)<-c('chrom','m6A','SNP','ref','alt','N','yi')
#Input_sample <- Input_sample[1:100,]     #for the test
```

```
##already given parameters
c <- 9
d <- 2
e <- 9
f <- 2
g <- 8
h <- 1.5
p <- 8
q <- 1.5
m <- 0.5
n <- 1

#Input_sample
pai<-results[1]
rho<-results[2]
sig<-results[3]
alp<-results[4]
beta<-results[5]
gam<-results[6]
lam<-results[7]

#getp<-function(Y=Input_sample[,7],N=Input_sample[,6]){
#getp<-function(Y,N){
Y=Input_sample[,7]
N=Input_sample[,6]
len<-length(Y)
Input_results<-data.frame(matrix(NA,n,10))
Input_results[,1:3]<-Input_sample[,1:3]
#Input_results<-Input_sample[,c(1:4,6:8)]       #use to record the p
test results
names(Input_results)<-
c('chrom','m6A','SNP','Input_null','Input_all','Input_p','Input_q','w
hesig','N','yi')
for(i in 1:len)
{
  nullValue<-pai/(pai+rho+sig)*pbb(Y[i],N[i],alp,alp)        #part  of
unbaised
  Input_results[i,4]<-nullValue
  allvalue<-
pai/(pai+rho+sig)*pbb(Y[i],N[i],alp,alp)+rho/(pai+rho+sig)*pbb(Y[i],N
[i],beta,gam)+sig/(pai+rho+sig)*ppois(Y[i],lam)
  Input_results[i,5]<-allvalue
```

```
  Input_results[i,6]<-nullValue/allvalue
}
Input_results[,7]<-p.adjust(Input_results[,6],method = 'fdr')
Input_results[,9]<-Input_sample$N
Input_results[,10]<-Input_sample$yi
for(i in 1:len)
{
  if (Input_results[i,6]<0.05){
    Input_results[i,8]<-'significant'
  } else {
      Input_results[i,8]<-'unsig'
      }
}
write.table(Input_results,file="Input_results.csv",sep =
",",row.names=F)
```

附录6 使用模拟数据集对模型进行检验

```
#use data to test the model
getwd()
setwd('C:/Users/Administrator/Desktop/m6A/R file')    #设置工作路径
.libPaths("C:/Program Files/R/R-3.4.4/library")    #set packages path

rm(list = ls())     #clear global environment

library(TailRank)

#timestart<-Sys.time()     #记录开始时间

file_name="test_results_0.2.csv"

num_vec<-vector(mode="numeric",length=100)

binomSample<-function(n,p=a){
  return(rbinom(1,n,p))
}
Pthetay<-function(yi,ni){      #here need to define all parameters in
advance
  partP<-
pai/(pai+rho+sig)*pbb(yi,ni,alp,alp)+rho/(pai+rho+sig)*pbb(yi,ni,beta
,gam)+sig/(pai+rho+sig)*ppois(yi,lam)
  return(log(partP))
}
```

```
for (j in 1:100){

#create biased data
#set.seed(3)
biasedN<-floor(runif(10,1,600))
a<-0.2
biasedbinom<-as.numeric(lapply(biasedN,binomSample))
biasedLabel<-rep("biased",10)
biasedSample<-data.frame(label=biasedLabel,yi=biasedbinom,N=biasedN)

#create unbiased data
#set.seed(13)
a<-0.5
unbiasedN<-floor(runif(90,1,600))
unbiasedbinom<-as.numeric(lapply(unbiasedN,binomSample))
unbiasedLabel<-rep("unbiased",90)
unbiasedSample<-
data.frame(label=unbiasedLabel,yi=unbiasedbinom,N=unbiasedN)
testSample<-rbind(unbiasedSample,biasedSample)
```