数据下载：

<https://github.com/liuxx1996/eQTLTutorial/blob/main/eQTLTutorial.zip>。

把下载得到的压缩包eQTLTutorial.zip解压，所使用的文件位于data/simulated中，包括一些模拟的数据集，所有数据的分隔符都是制表符（TAB）。共有四个模拟生成的数据，分别是两个基因表达数据（sim\_expression1.tab、sim\_expression2.tab）、基因型数据（sim\_genotypes.tab）和协变量数据（sim\_covariates.tab）。下面使用R分析单核苷酸多态性（SNP）与基因表达的关系。

本分析所用R节点信息如下：

R Code：

1. sessionInfo()

Return：

R version 3.6.3 (2020-02-29) #R版本

Platform: x86\_64-w64-mingw32/x64 (64-bit)

Running under: Windows 10 x64 (build 19042)

Matrix products: default

locale:

[1] LC\_COLLATE=Chinese (Simplified)\_China.936

[2] LC\_CTYPE=Chinese (Simplified)\_China.936

[3] LC\_MONETARY=Chinese (Simplified)\_China.936

[4] LC\_NUMERIC=C

[5] LC\_TIME=Chinese (Simplified)\_China.936

attached base packages: #基础包

[1] stats     graphics  grDevices utils     datasets  methods   base

loaded via a namespace (and not attached):

[1] compiler\_3.6.3 tools\_3.6.3

首先把工作目录更改到解压出的文件所在的文件夹下：

R Code：

1. setwd("K:/eQTLTutorial/") *#这里解压出文件位于的文件夹绝对路径为：K:/eQTLTutorial/。*

R Code：

1. dir() *#查看工作路径内的文件*

Return：

[1] "data"

然后安装分析过程所需要的R包：

R Code：

pkgs <- c("data.table","dplyr","ggplot2")

for(pkg in pkgs){

  if(!require(pkg,character.only = T))

    install.packages(pkg,repos="https://mirrors.tongji.edu.cn/CRAN/")

}

下面开始分析：

一．载入数据，可以使用RStudio的数据导入功能，或者使用命令行手动导入，本分析过程采用命令行方式导入：

RStudio：File 🡪 Import DataSet 🡪 From Text (text)，选择所要导入的文件，Heading选择Yes，Separator切换为Tab，取消勾选Strings as factors，Name按需修改。

命令行导入：

R Code：

1. genotype <- data.table::fread("data/simulated/sim\_genotypes.tab") *#导入基因型数据，并命名为genotype*

查看genotype的基本信息

R Code：

1. dim(genotype) *#查看genotype数据的维度*

Return：

[1] 300  11 *#共有300行（个体），11列（1列样本名称 + 10列SNP基因型）*

R Code：

1. head(genotype) *#查看genotype的前6行数据*

Return：

     sample snp\_1 snp\_2 snp\_3 snp\_4 snp\_5 snp\_6 snp\_7 snp\_8 snp\_9 snp\_10

1: sample\_1     0     0     0     0     1     0     0     0     2      2

2: sample\_2     0     0     0     0     0     0     0     0     2      2

3: sample\_3     0     0     0     0     0     0     0     0     1      2

4: sample\_4     0     0     0     0     0     0     1     1     0      1

5: sample\_5     0     0     0     0     0     0     0     1     0      1

6: sample\_6     0     1     0     0     0     0     0     0     0      0

*#一共有10个SNP位点，因为这里分析使用的是基因加性效应模型，SNP的基因型用0、1、2来编码，表示效应等位基因数量的多少。*

导入基因表达数据

R Code：

1. express1 <- data.table::fread("data/simulated/sim\_expression1.tab") *#导入基因表达数据，并命名为express1。*
2. dim(express1) *#查看数据维度*

Return：

[1] 300  11 *#表达数据一共有300行（个体），第一列是样本名称，后面10列是不同的基因。*

R Code：

1. head(express1) *#查看数据前6行。*

Return：

     sample   gene\_1   gene\_2   gene\_3   gene\_4   gene\_5   gene\_6   gene\_7

1: sample\_1 6.091035 6.479307 6.702659 7.260538 8.196299 6.061722 6.548776

2: sample\_2 7.432563 7.016138 7.508599 7.295998 7.836607 7.825705 7.514788

3: sample\_3 6.917247 6.127383 6.097665 6.324987 6.017944 5.888059 6.912573

4: sample\_4 7.318114 6.791866 7.942633 6.894796 7.498120 7.199753 7.900366

5: sample\_5 7.288953 6.982799 8.120189 7.400018 7.401130 7.032545 7.413375

6: sample\_6 7.688428 8.420471 7.709514 8.422695 6.536603 8.340534 8.668130

     gene\_8    gene\_9   gene\_10

1: 6.316635  9.646378  9.811172

2: 8.359249 10.433948 10.455313

3: 6.830853  8.312266  9.678442

4: 8.651775  6.560307  9.509531

5: 8.948665  7.482829  8.749063

6: 7.582129  7.638559  8.011134

需要检查基因型和表达数据中每一行的样本是否相同：

R Code：

1. all(genotype$sample==express1$sample)

Return：

[1] TRUE *#两个数据样本一一对应*

二．计算次等位基因频率。因为我们的分析基于加性效应模型，SNP基因型被编码为每个个体所携带的第二个等位基因的拷贝数，在eQTL分析中，有必要确保第二个等位基因对应于低频的次等位基因（minor allele），这样分析过程的效应等位基因是低频的等位基因，将有助于后续对基因型效应的解释。为了保证这一前提，首先查看基因型数据中默认的效应等位基因的频率：

R Code：

1. esnpf <- genotype[,-1][,lapply(.SD, function(x){sum(x)/(2\*.N)})] *#每个SNP的效应等位基因频率为效应等位基因个数之和除以等位基因总个数（人数两倍）*
2. esnpf

Return：

         snp\_1 snp\_2 snp\_3 snp\_4      snp\_5     snp\_6     snp\_7     snp\_8

1: 0.006666667  0.03  0.02 0.065 0.04666667 0.1266667 0.1716667 0.2983333

   snp\_9    snp\_10

1:  0.39 0.5116667

找出频率大于0.5的：

R Code：

1. esnpf[,unlist(esnpf) > 0.5,with=F]

Return：

      snp\_10

1: 0.5116667

结果表明，只有snp\_10的基因型表示的是高频等位基因（称作主等位基因，major allele），为了保证我们后面使用的是次等位基因进行分析，需要将该SNP倒置转换：

R Code：

1. genotype[,snp\_10:=as.integer(dplyr::case\_when(snp\_10==2~0,snp\_10==0~2,T~1))]*#将2重新编码为0，0重新编码为2，1不变*

重新计算频率：

esnpf <- genotype[,-1][,lapply(.SD, function(x){sum(x)/(2\*.N)})]

esnpf

Return：

         snp\_1 snp\_2 snp\_3 snp\_4      snp\_5     snp\_6     snp\_7     snp\_8

1: 0.006666667  0.03  0.02 0.065 0.04666667 0.1266667 0.1716667 0.2983333

   snp\_9    snp\_10

1:  0.39 0.4883333

可以看到snp\_10的基因型频率已经小于0.5。

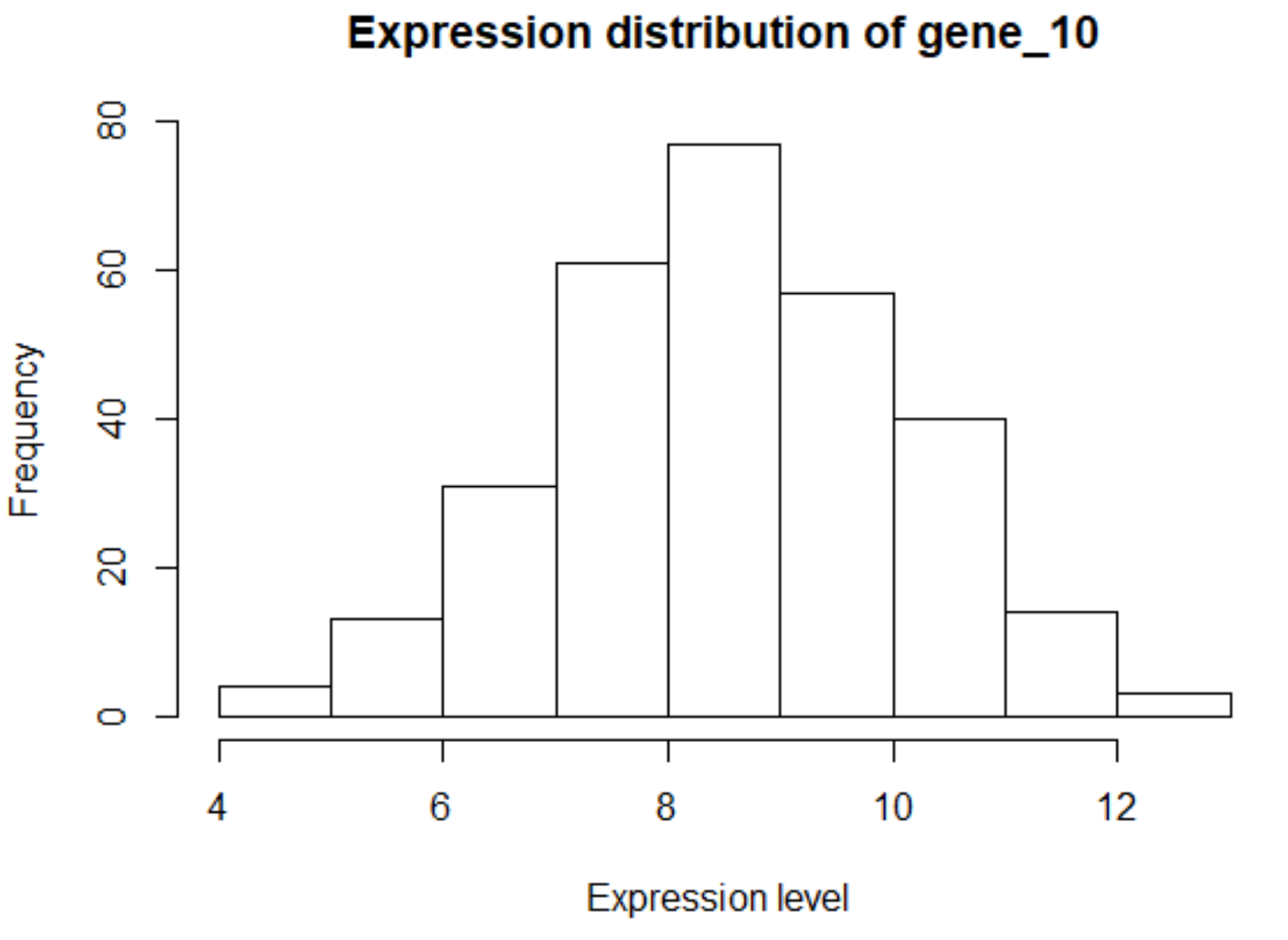
三．检查基因表达数据。我们已经了解了基因型数据所包含的信息，下面检查一下基因表达数据，一般而言，在eQTL分析中，基因表达数据应该在不同样本中近似正态分布，否则应进行相应的处理，比如去除离群值、标准正态转化等。下面，我们检查一下本次所分析的基因表达数据的正态性：

直观的方式是绘制每个基因表达的分布直方图（或密度图），来查看分布是否呈现偏态，显著偏离正态分布的钟形曲线，以基因gene\_10为例：

R Code：

1. hist(express1$gene\_10,main="Expression distribution of gene\_10",xlab="Expression level ") *#绘制gene\_10表达水平的直方图*

Return:



结果图可以看出，gene\_10的表达接近正态分布。

这种检查方式虽然简单直观，但是缺乏准确性，而且难以批量查看所有基因的情况，下面使用Shapiro-Wilk检验10个基因的正态性：

R Code：

1. expnorm <- express1[,-1][,lapply(.SD,function(x)shapiro.test(x)$p.value)] *#使用Shapiro-Wilk正态性检验，对所有基因表达的分布情况进行检测*
2. expnorm

Return：

      gene\_1    gene\_2    gene\_3    gene\_4    gene\_5    gene\_6    gene\_7

1: 0.3656916 0.4171927 0.5880415 0.3812846 0.4251698 0.5689382 0.1329803

     gene\_8     gene\_9   gene\_10

1: 0.346955 0.01143871 0.7394773

R Code：

1. expnorm[,unlist(expnorm <0.01),with=F]

Return：

Null data.table (0 rows and 0 cols)

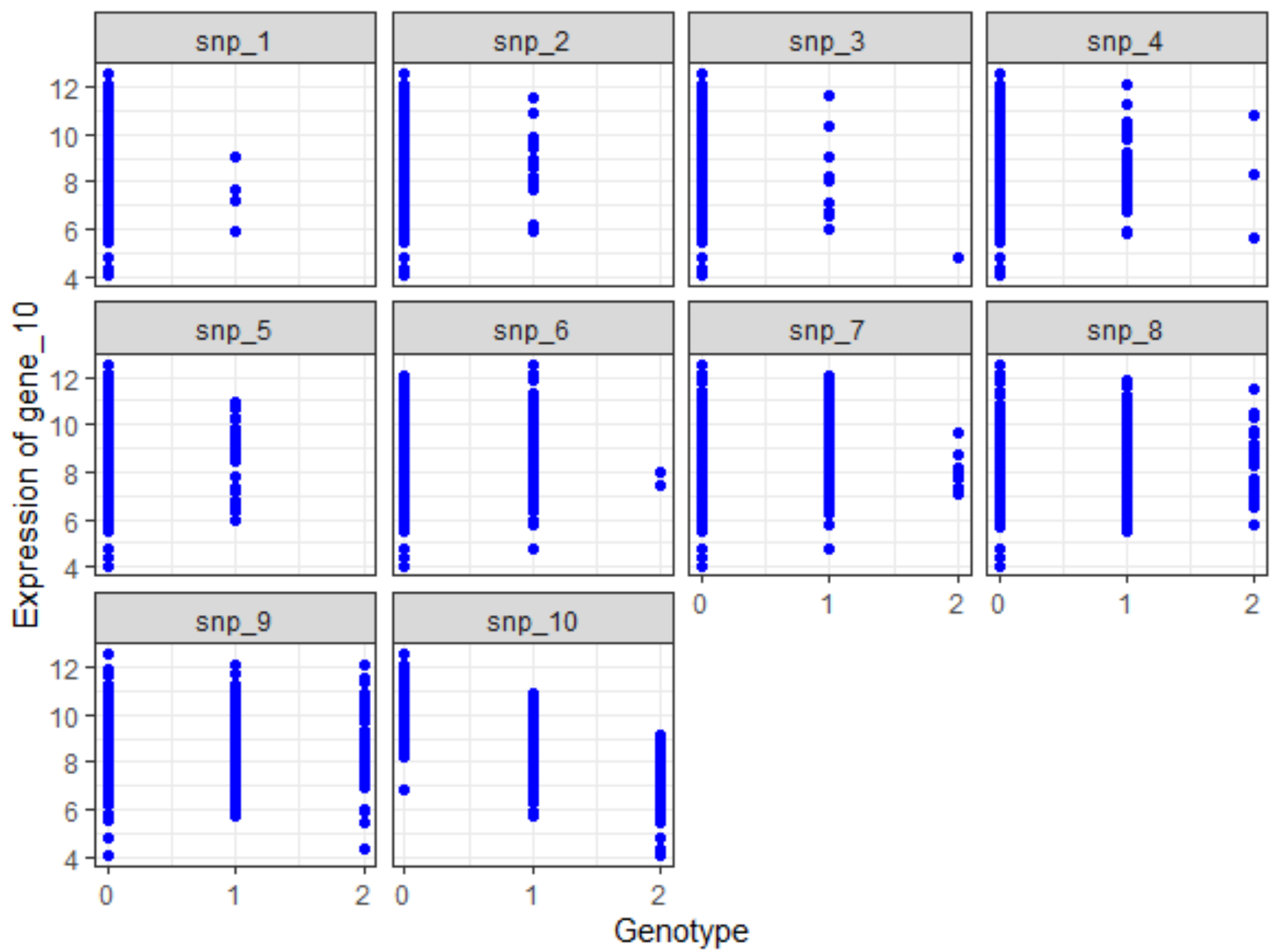
没有基因表达的分布显著偏离正态分布。

四．检查不同基因型与基因表达的关系图。为了分析不同基因型对基因表达的影响，使用散点图绘制不同基因型对应的基因表达数据可以直观的展现我们想要的结果。下面以gene\_10为例，进行展示：

R Code：

1. library(ggplot2) *#使用ggplot2绘制*
2. gene10\_snp=cbind(genotype[,-1],express1[,"gene\_10"]) *#gene\_10与所有SNP数据*
3. plotdata = data.table::melt(gene10\_snp,id.vars="gene\_10")
4. ggplot(data = plotdata, aes(value, gene\_10)) +
5. geom\_point(color = "blue") + *#绘制点*
6. scale\_x\_continuous(breaks = c(0,1,2)) + *#x轴坐标，表示基因型*
7. facet\_wrap(~variable) + *#按SNP划分不同图*
8. labs(x = "Genotype",y = "Expression of gene\_10") +
9. theme\_bw() *#修改图背景*

Return：

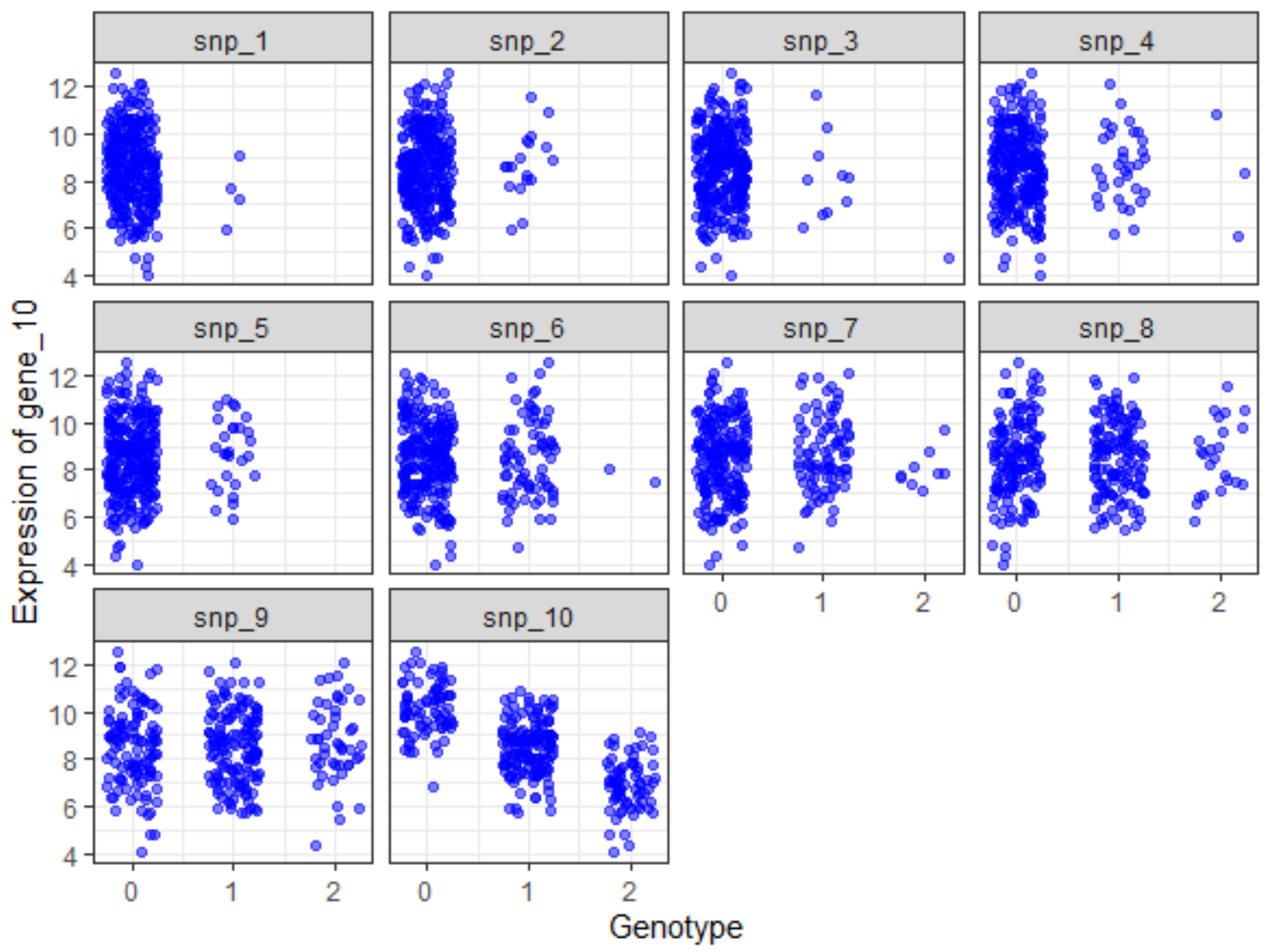


上面10张图分别展示了10个SNP与gene\_10表达水平的关系，不过很多点重合到一起，看起来并不清晰，可以给每个点加上一些扰动，并把颜色透明度调低，使图像看起来更明了：

R Code：

1. ggplot(data = plotdata, aes(value, gene\_10)) +
2. geom\_jitter(color="blue",alpha = 0.5,position=position\_jitter(0.25)) + #扰动
3. scale\_x\_continuous(breaks = c(0,1,2)) +
4. facet\_wrap(~variable) +
5. labs(x = "Genotype",y = "Expression of gene\_10") +
6. theme\_bw()

Return：



可以看到，前9个SNP基因型与gene\_10的表达水平看起来关系不大，而snp\_10不同基因型很可能造成gene\_10表达水平的差异。

五．估计SNP的效应。为了估计每个SNP对基因表达的影响，我们将使用简单线性回归模型进行拟合：，其中，是基因在所有样本中的表达值向量，是SNP在所有样本中的基因型向量。我们主要关注并估计它的值，它表示了SNP的次等位基因每个拷贝造成基因表达水平的变化。下面将使用线性回归分析每一个SNP基因型对gene\_10表达水平的效应大小。

R Code：

1. gene10fit <- lapply(genotype[,-1],function(x){lm(express1$gene\_10~x)}) *#将gene\_10表达水平作为因变量，使用每个SNP基因型进行简单线性回归，得到每个拟合的lm模型。*
2. gene10fit$snp\_1 *#查看snp\_1对gene\_10的影响大小*

Return：

Call:

lm(formula = express1$gene\_10 ~ x)

Coefficients:

(Intercept)            x

      8.553       -1.078

#结果中，Coefficients里x的值-1.078就是线性拟合估计的斜率，在eQTL分析中，叫做效应大小（effect size）或beta。

更为简单的方法是使用coef()来获得每一个SNP的效应估计：

R Code：

1. ef <- sapply(gene10fit, coef)["x",]
2. ef

Return：

      snp\_1       snp\_2       snp\_3       snp\_4       snp\_5

-1.07768367  0.17537415 -0.81179058  0.04799428  0.23233727

      snp\_6       snp\_7       snp\_8       snp\_9      snp\_10

-0.06313759  0.08069206  0.03390471  0.11187799 -1.61627302

除了知道SNP的效应大小估计值之外，我们还可以获得每个SNP效应的95%置信区间：

R Code：

1. confint(gene10fit$snp\_1, "x") *#snp\_1的置信区间*

Return：

     2.5 %    97.5 %

x -2.63508 0.4797123

R Code：

1. gene10ci <- sapply(gene10fit, confint, "x") *#所有SNP的效应置信区间*
2. rownames(gene10ci) <- c("2.5 %", "97.5 %")
3. gene10ci

Return：

            snp\_1      snp\_2       snp\_3      snp\_4      snp\_5

2.5 %  -2.6350797 -0.5788643 -1.65076131 -0.4431225 -0.3830616

97.5 %  0.4797123  0.9296126  0.02718015  0.5391111  0.8477361

            snp\_6      snp\_7      snp\_8      snp\_9    snp\_10

2.5 %  -0.4612715 -0.2545574 -0.2515380 -0.1433647 -1.789212

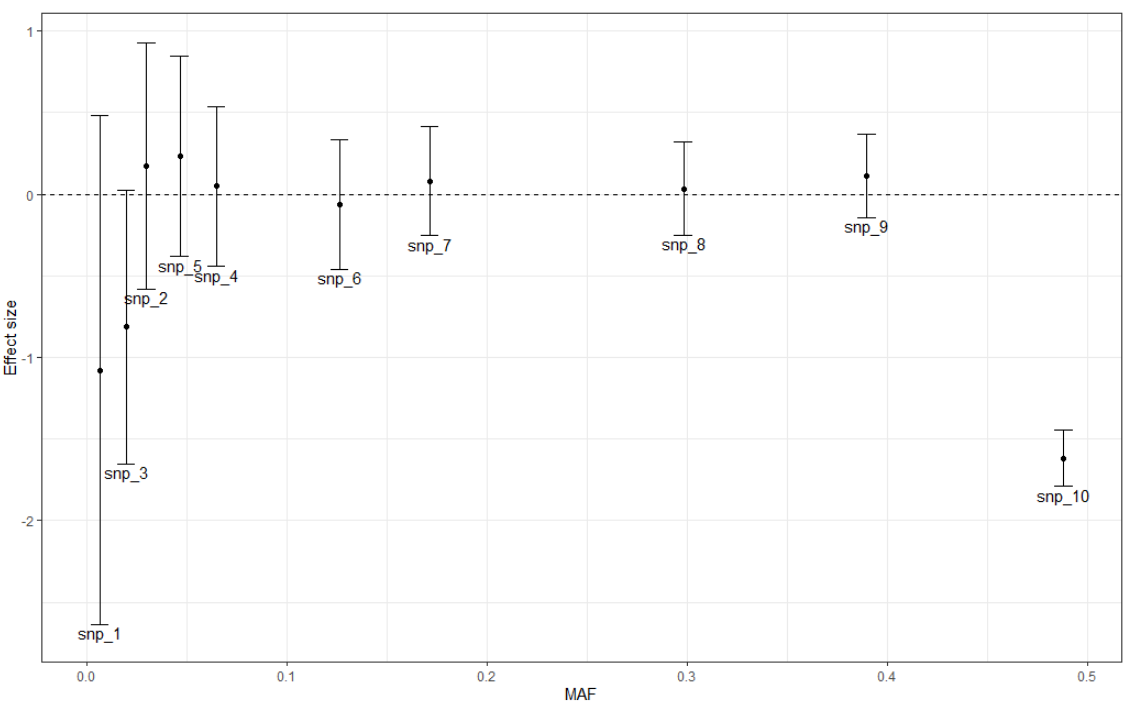
97.5 %  0.3349964  0.4159415  0.3193474  0.3671207 -1.443334

进一步，我们可以绘制图形来展示每个SNP的效应大小：

R Code：

1. plotdata2 <- data.frame(SNP=colnames(esnpf),MAF=unlist(esnpf),
2. ef,l=gene10ci[1,],u=gene10ci[2,])
3. ggplot(plotdata2, aes(x=MAF))+ #以每个SNP的效应等位基因频率为横轴
4. geom\_errorbar(aes(ymin=l, ymax=u)) + #效应置信区间
5. geom\_hline(yintercept = 0,linetype="dashed") + #参考线
6. geom\_point(aes(y=ef)) +
7. geom\_text(aes(y=l-.05,label=SNP)) +
8. labs(y = "Effect size") +
9. theme\_bw()

Return：



结果显示，除了snp\_10，其他9个SNP基因型对gene\_10表达水平的效应95%置信区间都包括0效应。置信区间的长度可以反映估计效应的不确定性，从图中可以看出，随着等位基因频率的增加，这种不确定性逐渐降低，不过在MAF很高的情况下，模型仍然存在一定的不确定性。

六．分析SNP与基因表达关联显著性。除了了解每个SNP基因型对基因表达的影响大小之外，我们通常还想知道哪些变量是重要的。利用五中得到的拟合模型，可以很方便地计算出每个SNP对基因表达水平影响的显著性大小。

R Code：

1. summary(gene10fit$snp\_1) *#使用lm模型的summary方法，查看gene\_10和snp\_1之间线性回归模型各系数的显著性*

Return：

Call:

lm(formula = express1$gene\_10 ~ x)

Residuals:

    Min      1Q  Median      3Q     Max

-4.5262 -1.1412  0.0577  1.1552  4.0071

Coefficients:

            Estimate Std. Error t value Pr(>|t|)

(Intercept)  8.55324    0.09138  93.600   <2e-16 \*\*\*

x           -1.07768    0.79138  -1.362    0.174

---

Signif. codes:  0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.572 on 298 degrees of freedom

Multiple R-squared:  0.006185, Adjusted R-squared:  0.00285

F-statistic: 1.854 on 1 and 298 DF,  p-value: 0.1743

summary返回了模型拟合的全面结果，因为我们在研究eQTL，所以主要关注的内容是snp\_1对gene\_10表达水平的影响，这一部分内容包含在系数表（Coefficients）里，这个表格中展示了我们已看到过的估计系数（第一列），第二、三列分别是系数的标准误和t检验统计量（等于估计系数与标准误之商），第四列给出了根据t统计量计算的p值，第五列给出了一个星级评定：所有p值小于0.05的得到一星，小于0.01的得到两颗星，以此类推，这使得它便于快速查看哪些变量有显著效果。在本结果中，系数表的第一行是截距的数据，第二行是snp\_1的，可以看到，snp\_1基因型与gene\_10的表达水平之间关联并不显著。

在eQTL研究中，比较常见的是从成千上万个SNP中找出与一系列基因的表达相关联的位点，最常见的途径是通过它们的p值来鉴定eQTL。下面我们将计算所有SNP对gene\_10表达水平的关联显著性p值，以进行比较。

R Code：

1. coeftable <- sapply(gene10fit,function(x){summary(x)$coef["x",]}) *#计算所有SNP与gene\_10表达水平线性拟合模型的SNP效应表*
2. coeftable

Return：

                snp\_1     snp\_2       snp\_3      snp\_4     snp\_5

Estimate   -1.0776837 0.1753742 -0.81179058 0.04799428 0.2323373

Std. Error  0.7913772 0.3832597  0.42631569 0.24955673 0.3127096

t value    -1.3617825 0.4575856 -1.90420059 0.19231810 0.7429810

Pr(>|t|)    0.1742951 0.6475836  0.05784699 0.84762403 0.4580785

                 snp\_6      snp\_7      snp\_8     snp\_9

Estimate   -0.06313759 0.08069206 0.03390471 0.1118780

Std. Error  0.20230831 0.17035410 0.14504524 0.1296994

t value    -0.31208601 0.47367252 0.23375268 0.8625948

Pr(>|t|)    0.75519341 0.63608028 0.81533748 0.3890539

                  snp\_10

Estimate   -1.616273e+00

Std. Error  8.787721e-02

t value    -1.839240e+01

Pr(>|t|)    5.187685e-51

*#这里包含了每一个SNP的效应结果，其中第四行是p值*

R Code：

1. pval <- coeftable[4,] *#系数表第四行就是每个SNP的显著性*
2. pval

Return：

       snp\_1        snp\_2        snp\_3        snp\_4        snp\_5

1.742951e-01 6.475836e-01 5.784699e-02 8.476240e-01 4.580785e-01

       snp\_6        snp\_7        snp\_8        snp\_9       snp\_10

7.551934e-01 6.360803e-01 8.153375e-01 3.890539e-01 5.187685e-51

R Code：

1. pval[pval<0.05] #显著性水平小于0.05的

Return：

     snp\_10

5.187685e-51

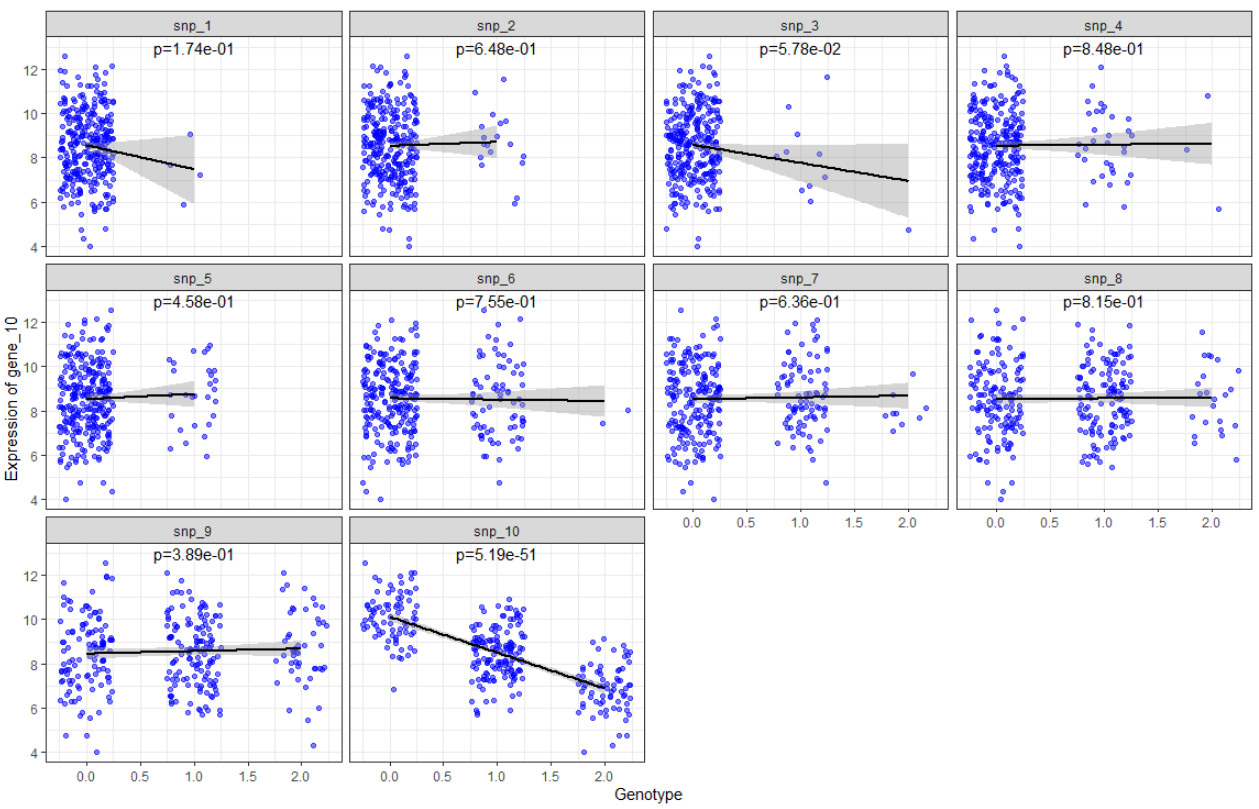
*#只有snp\_10*

我们可以将数据点、拟合直线以及显著性共同展示出来。

R Code：

1. textdata=data.frame(variable=names(pval),x=1,y=13,
2. label=formatC(pval,digits=2,format="e")) #p值
3. ggplot(data=plotdata,aes(x=value,y=gene\_10))+
4. geom\_jitter(color="blue",alpha = 0.5,position=position\_jitter(0.25))+ #添加点
5. geom\_smooth(method="lm",se=T,color="black",formula=y~x) + #添加拟合直线
6. geom\_text(data=textdata,aes(x=x,y=y,
7. label=paste0("p=",label)))+ #添加p值
8. facet\_wrap(~variable)+
9. labs(x = "Genotype",y = "Expression of gene\_10") +
10. theme\_bw()

Return：

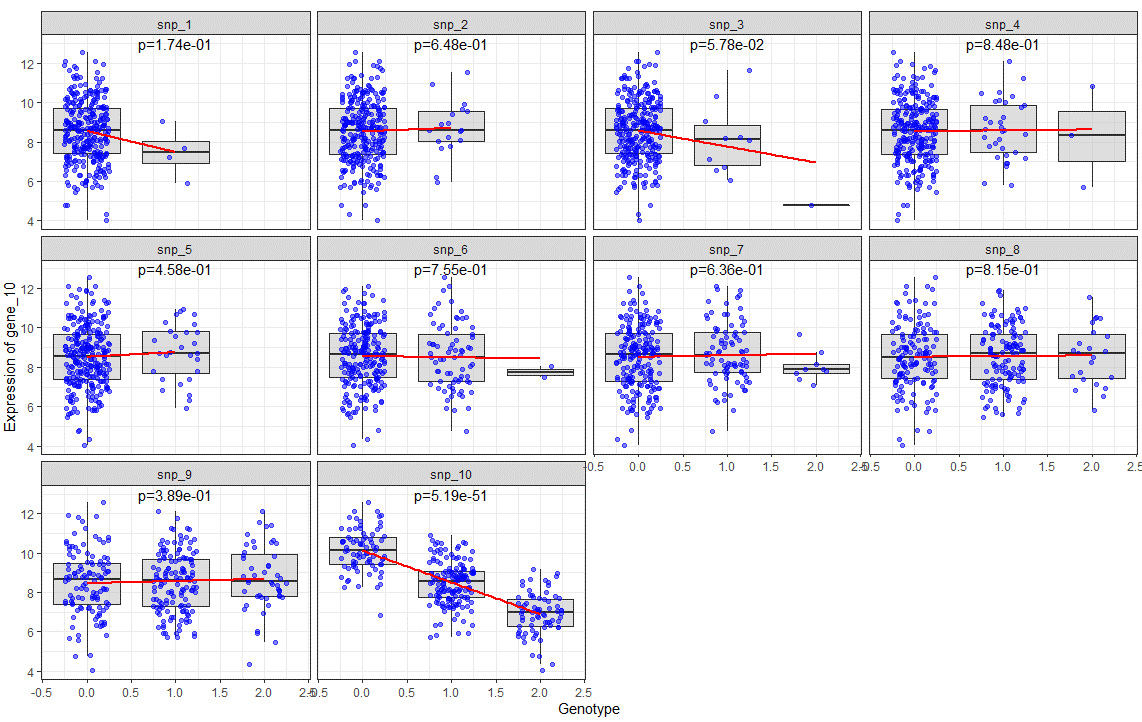


使用散点来展示的效果并不是太好，即使加上了一些扰动，仍然显得比较乱，尤其是当数据点很多的时候，这种情况下，使用箱线图来展示会更直观一些。

R Code：

1. ggplot(data=plotdata,aes(x=value,y=gene\_10))+
2. geom\_boxplot(aes(group=value),outlier.shape=NA,fill="gray",alpha=0.5) + #添加箱线图
3. geom\_jitter(color="blue",alpha = 0.5,position=position\_jitter(0.25))+
4. geom\_smooth(method="lm",se=F,color="red",formula=y~x) +
5. geom\_text(data=textdata,aes(x=x,y=y,
6. label=paste0("p=",label)))+
7. facet\_wrap(~variable)+
8. labs(x = "Genotype",y = "Expression of gene\_10") +
9. theme\_bw()

Return：



七．添加协变量以排除一些潜在的干扰因素。许多不同的因素都可能影响基因表达水平，比如年龄、性别、吸烟行为、基因突变、营养条件等。在模型中加入协变量共同分析，可以使结果更加准确，而且也能提高找到更多微效遗传因子的可能性。

协变量是用来描述影响基因表达的样本特征，技术上来说，一个协变量数目和样本数目相同，比如年龄。之前的例子分析比较顺利，主要是因为使用的模拟数据没有添加协变量的干扰，现在，我们将使用一组加入了协变量的数据来进行分析，比较模型中协变量添加与否对结果显著性的影响，并绘图展示结果。

1.载入数据：

R Code：

1. express2 <- data.table::fread("data/simulated/sim\_expression2.tab") *#新的表达数据*
2. all(genotype$sample==express2$sample) *#检查表达数据和基因型数据中每一行的样本是否相同*

Return：

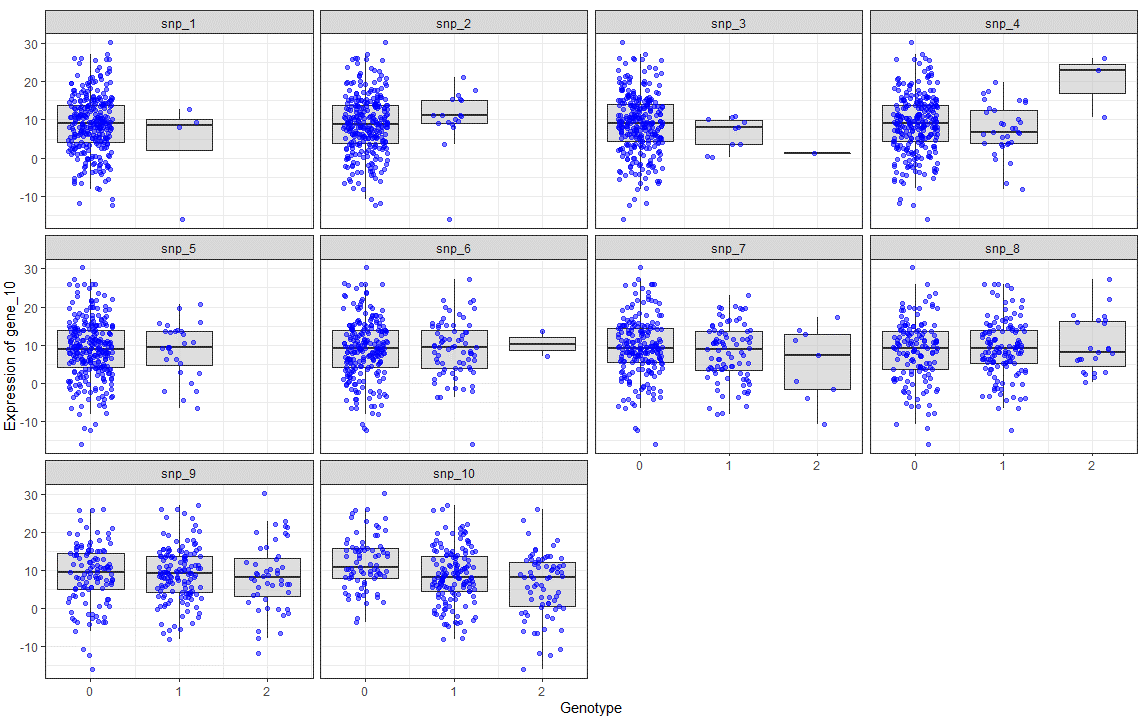
[1] TRUE *#样本一一对应*

2.同样的，我们以gene\_10为例。直接绘制SNP不同基因型和gene\_10表达水平的关系。

R Code：

1. gene10\_snp2=cbind(genotype[,-1],express2[,"gene\_10"]) *#gene\_10与所有SNP数据*
2. plotdata3 = data.table::melt(gene10\_snp2,id.vars="gene\_10")
3. ggplot(data = plotdata3, aes(value, gene\_10)) +
4. geom\_boxplot(aes(group=value),outlier.shape=NA,fill="gray",alpha=0.5) +
5. geom\_jitter(color="blue",alpha = 0.5,position=position\_jitter(0.25)) + *#扰动*
6. scale\_x\_continuous(breaks = c(0,1,2)) +
7. facet\_wrap(~variable) +
8. labs(x = "Genotype",y = "Expression of gene\_10") +
9. theme\_bw()

Return：



下面，我们在回归中进一步分析它们的关系，像之前一样，找出各SNP效应及95%置信区间。

R Code：

1. gene10fit2 <- lapply(genotype[,-1],function(x){lm(express2$gene\_10~x)}) *#线性拟合模型*
2. ef2 <- sapply(gene10fit2, coef)["x",] *#每个SNP的效应估计*
3. ef2

Return：

     snp\_1      snp\_2      snp\_3      snp\_4      snp\_5

-5.4207548  1.7008595 -2.8614062  0.6010741 -0.2384864

     snp\_6      snp\_7      snp\_8      snp\_9     snp\_10

 0.1820607 -1.5977234  0.7441472 -0.1192989 -2.9264897

R Code：

1. gene10ci2 <- sapply(gene10fit2, confint, "x") *#SNP效应95%置信区间*
2. rownames(gene10ci2) <- c("2.5 %", "97.5 %")
3. gene10ci2

Return：

            snp\_1     snp\_2     snp\_3     snp\_4     snp\_5

2.5 %  -13.001750 -1.967514 -6.957961 -1.789221 -3.237365

97.5 %   2.160241  5.369233  1.235148  2.991369  2.760392

           snp\_6       snp\_7      snp\_8     snp\_9    snp\_10

2.5 %  -1.756563 -3.22039584 -0.6431404 -1.363490 -4.110736

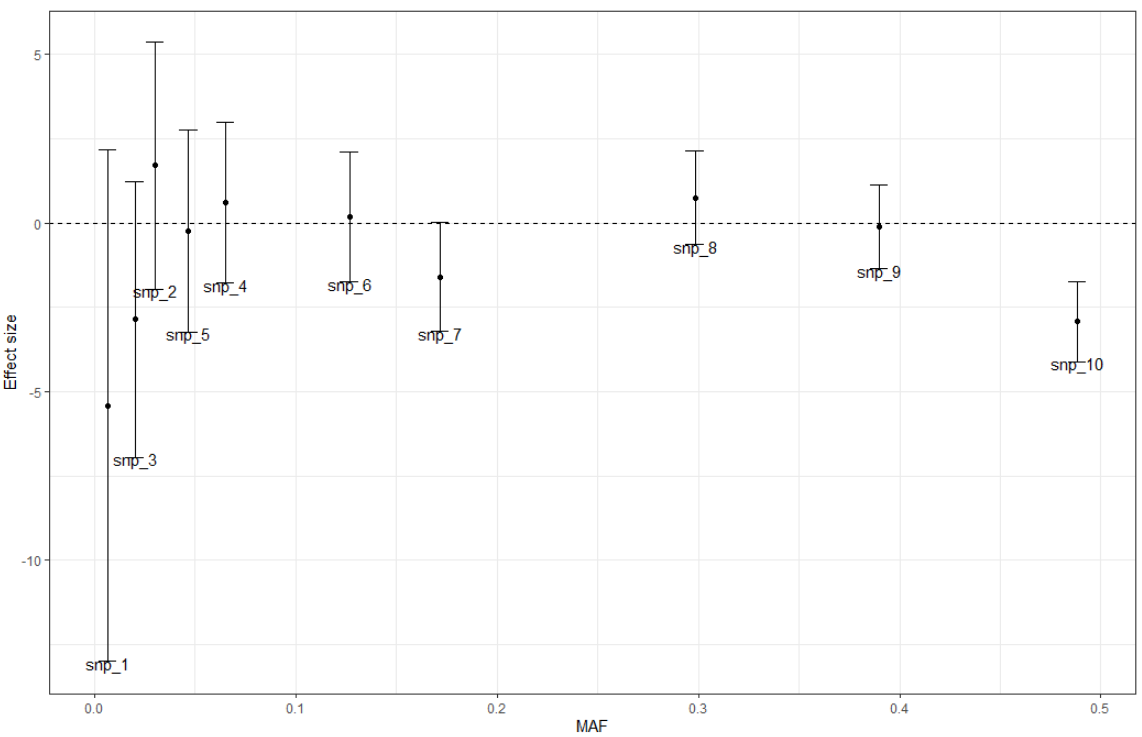
97.5 %  2.120684  0.02494895  2.1314348  1.124892 -1.742243

绘图加以展示：

R Code：

1. plotdata4 <- data.frame(SNP=colnames(esnpf),MAF=unlist(esnpf),
2. ef2,l=gene10ci2[1,],u=gene10ci2[2,])
3. ggplot(plotdata4, aes(x=MAF))+ #各SNP的效应等位基因频率为横轴
4. geom\_errorbar(aes(ymin=l, ymax=u)) + #效应置信区间
5. geom\_hline(yintercept = 0,linetype="dashed") + #参考线
6. geom\_point(aes(y=ef2)) + #效应大小估计值
7. geom\_text(aes(y=l-.05,label=SNP)) +
8. labs(y = "Effect size") +
9. theme\_bw()

Return：



这次得到的置信区间比之前大很多。我们继续计算显著性。

R Code：

1. coeftable2 <- sapply(gene10fit2,function(x){summary(x)$coef["x",]})
2. pval2 <- coeftable2[4,] *#每个SNP的显著性*
3. pval2

Return：

      snp\_1        snp\_2        snp\_3        snp\_4        snp\_5

1.604168e-01 3.622680e-01 1.702883e-01 6.210559e-01 8.757432e-01

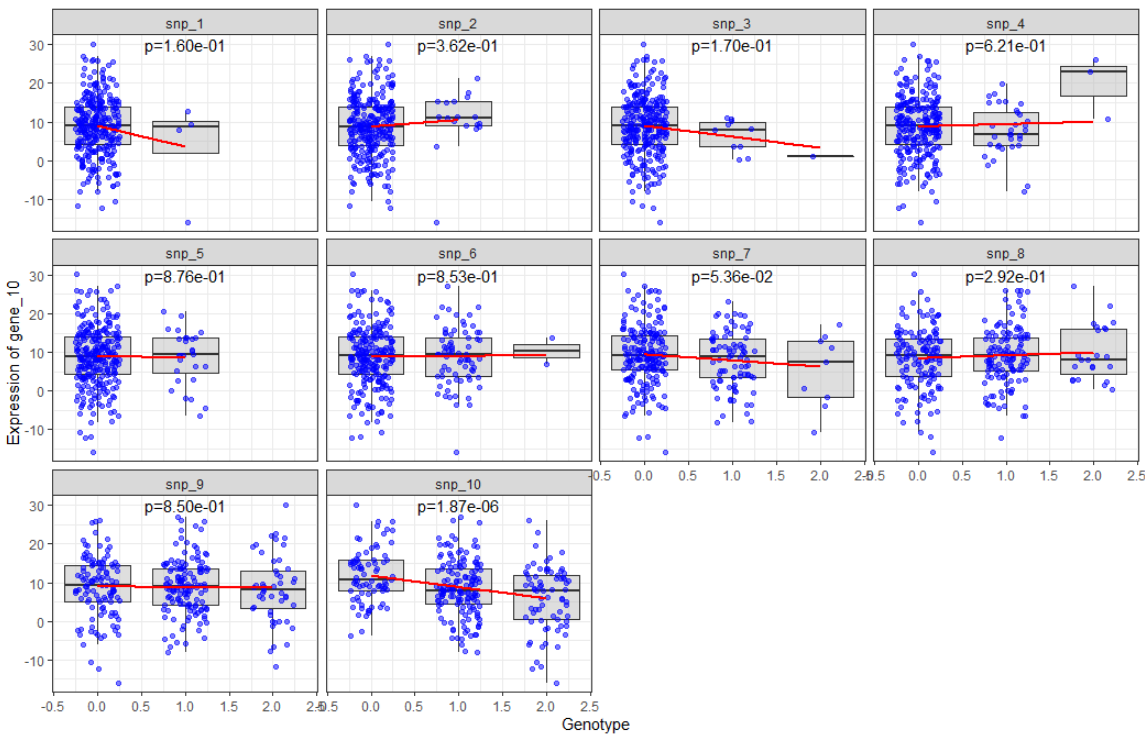
       snp\_6        snp\_7        snp\_8        snp\_9       snp\_10

8.534995e-01 5.360456e-02 2.919967e-01 8.504588e-01 1.873111e-06

R Code：

1. textdata2=data.frame(variable=names(pval2),label=formatC(pval2,digits=2,format="e"))
2. ggplot(data=plotdata3,aes(x=value,y=gene\_10))+
3. geom\_boxplot(aes(group=value),outlier.shape=NA,fill="gray",alpha=0.5) +
4. geom\_jitter(color="blue",alpha = 0.5,position=position\_jitter(0.25))+
5. geom\_smooth(method="lm",se=F,color="red",formula=y~x) +
6. geom\_text(data=textdata2,aes(x=1,y=30,
7. label=paste0("p=",label)))+
8. facet\_wrap(~variable)+
9. labs(x = "Genotype",y = "Expression of gene\_10") +
10. theme\_bw()

Return：



接下来，我们把协变量纳入分析模型中。首先只纳入一部分协变量。

R Code：

1. covariates <- data.table::fread("data/simulated/sim\_covariates.tab") *#协变量数据*
2. all(genotype$sample==covariates$sample) *#检查协变量和基因型数据中每一行样本是否一一对应*

Return：

[1] TRUE *#样本相同*

R Code：

1. head(covariates,2)

Return：

     sample     var\_1     var\_2     var\_3     var\_4      var\_5

1: sample\_1 0.1805229 0.6573043 2.1202020 0.7086180  0.2261268

2: sample\_2 0.7847340 1.0363177 0.3571627 0.8541129 -0.3856621

       var\_6     var\_7     var\_8     var\_9    var\_10      var\_11

1: -1.323359 0.0693342 -1.270093 1.9314654 0.1893478 -0.02579421

2:  1.015386 0.9291073  0.790172 0.7504875 0.1566840  0.46778575

      var\_12     var\_13     var\_14    var\_15     var\_16

1: 1.4651498 -1.7309505 -0.8570105 0.3502420 -0.5500488

2: 0.4504485  0.2793073 -0.4621151 0.8739229 -0.9148169

      var\_17    var\_18     var\_19      var\_20

1: 0.5100173 -1.728002  0.6182252 -0.09250967

2: 1.0457920 -1.032430 -3.0233684 -0.16821972

*#一共有20个协变量*

R Code：

1. gene10fit3<-lapply(genotype[,-1],function(x){
2. lm(express2$gene\_10~x+as.matrix(covariates[,2:6]))}) *#只加入前5个协变量*
3. ef3 <- sapply(gene10fit3, coef)["x",]
4. ef3 *#SNP效应估计值*

Return：

    snp\_1       snp\_2       snp\_3       snp\_4       snp\_5

-2.90788677  1.38424323 -1.89324120  0.78811968 -1.22285913

      snp\_6       snp\_7       snp\_8       snp\_9      snp\_10

 1.08605219 -1.28618653 -0.05258846  0.15171078 -1.89560493

R Code：

1. gene10ci3 <- sapply(gene10fit3, confint, "x") *#SNP效应95%置信区间*
2. rownames(gene10ci3) <- c("2.5 %", "97.5 %")
3. gene10ci3

Return：

          snp\_1     snp\_2     snp\_3      snp\_4      snp\_5

2.5 %  -8.491333 -1.301985 -4.894375 -0.9893946 -3.4068179

97.5 %  2.675559  4.070472  1.107893  2.5656339  0.9610996

            snp\_6       snp\_7      snp\_8      snp\_9     snp\_10

2.5 %  -0.3274641 -2.47368470 -1.0759328 -0.7596326 -2.7938167

97.5 %  2.4995685 -0.09868835  0.9707558  1.0630542 -0.9973932

R Code：

plotdata5 <- data.frame(SNP=colnames(esnpf),MAF=unlist(esnpf),

                        ef3,l=gene10ci3[1,],u=gene10ci3[2,])

ggplot(plotdata5, aes(x=MAF))+ #SNP的效应等位基因频率

  geom\_errorbar(aes(ymin=l, ymax=u)) + #效应置信区间

  geom\_hline(yintercept = 0,linetype="dashed") +

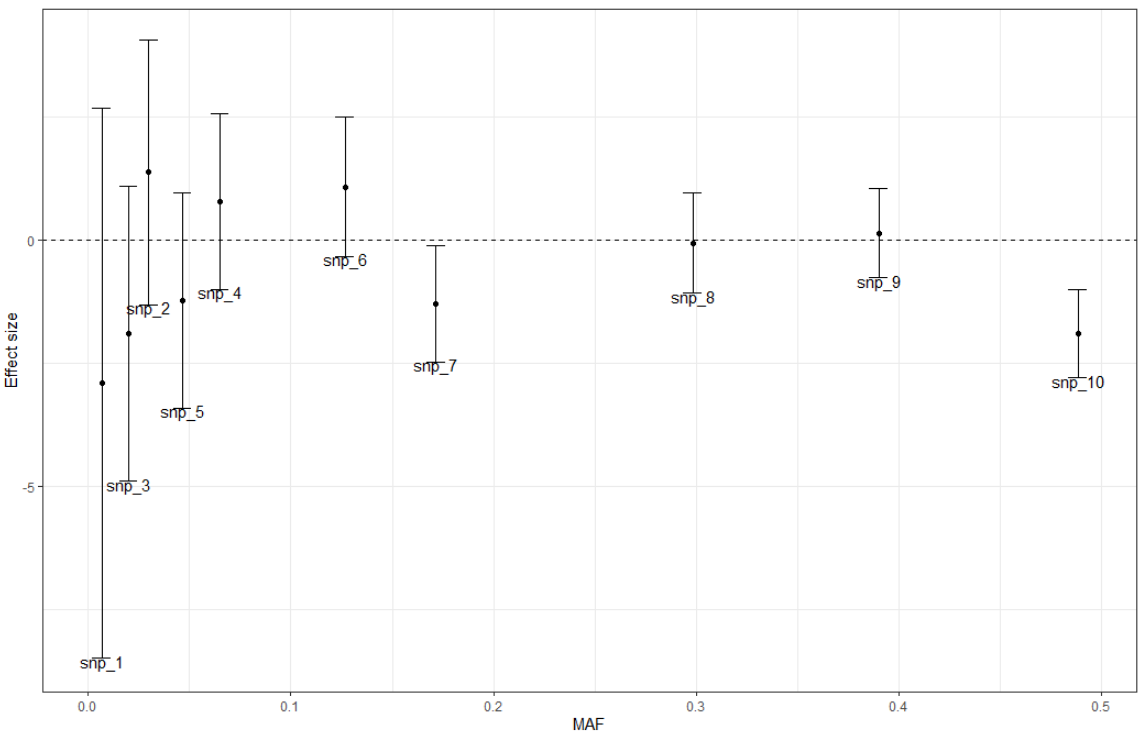
  geom\_point(aes(y=ef3)) + #效应大小估计值

  geom\_text(aes(y=l-.05,label=SNP)) +

  labs(y = "Effect size") +

  theme\_bw()

Return：



与不加入协变量相比，这次结果的置信区间范围缩小，更加紧密。

最后，我们把所有协变量全部纳入模型中，分析过程与之前类似。

R Code：

1. gene10fit4<-lapply(genotype[,-1],function(x){
2. lm(express2$gene\_10~x+as.matrix(covariates[,-1]))})
3. ef4 <- sapply(gene10fit4, coef)["x",]
4. ef4 *#SNP效应大小*

Return：

     snp\_1       snp\_2       snp\_3       snp\_4       snp\_5

-1.29223842  0.13758652 -0.76646202 -0.11735251  0.18089802

      snp\_6       snp\_7       snp\_8       snp\_9      snp\_10

-0.03920119  0.10379335  0.03135280  0.07679751 -1.65775585

R Code：

1. gene10ci4 <- sapply(gene10fit4, confint, "x") *#SNP效应95%置信区间*
2. rownames(gene10ci4) <- c("2.5 %", "97.5 %")
3. gene10ci4

Return：

            snp\_1      snp\_2      snp\_3      snp\_4      snp\_5

2.5 %  -2.9399822 -0.6504013 -1.6359944 -0.6338454 -0.4577176

97.5 %  0.3555053  0.9255744  0.1030703  0.3991404  0.8195136

            snp\_6      snp\_7      snp\_8      snp\_9    snp\_10

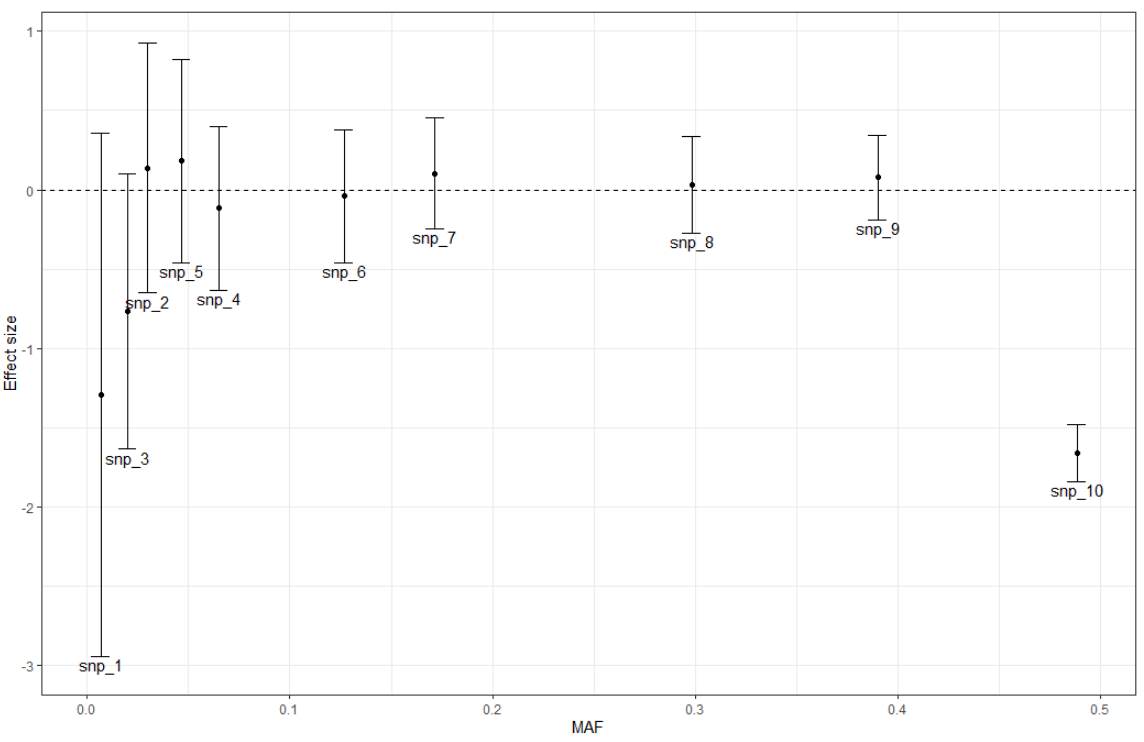
2.5 %  -0.4593554 -0.2477400 -0.2704726 -0.1912294 -1.838309

97.5 %  0.3809530  0.4553267  0.3331782  0.3448245 -1.477203

R Code：

1. plotdata6 <- data.frame(SNP=colnames(esnpf),MAF=unlist(esnpf),
2. ef4,l=gene10ci4[1,],u=gene10ci4[2,])
3. ggplot(plotdata6, aes(x=MAF))+ #将SNP的效应等位基因频率作为横轴
4. geom\_errorbar(aes(ymin=l, ymax=u)) + #效应置信区间
5. geom\_hline(yintercept = 0,linetype="dashed") + #参考线
6. geom\_point(aes(y=ef4)) + #效应大小估计值
7. geom\_text(aes(y=l-.05,label=SNP)) +
8. labs(y = "Effect size") +
9. theme\_bw()

Return：



在分析模型中包含全部的协变量后，得到的结果与最开始的简单示例中产生的结果相似，这表明如果能考虑到所有的混杂因素，基因型效应可以恢复。下面我们对比一下显著性的变化。

R Code：

1. coeftable3 <- sapply(gene10fit4,function(x){summary(x)$coef["x",]})
2. pval3 <- coeftable3[4,] *#每个SNP的显著性*
3. pval3

Return：

       snp\_1        snp\_2        snp\_3        snp\_4        snp\_5

1.237701e-01 7.313200e-01 8.381482e-02 6.550275e-01 5.775536e-01

       snp\_6        snp\_7        snp\_8        snp\_9       snp\_10

8.544078e-01 5.615583e-01 8.381234e-01 5.731801e-01 7.986547e-49

与之前结果对比可以看到，显著性p值同样得到了恢复。

将所有协变量纳入分析模型的方法直观简单，但如果要分析的SNP数目非常多，分析时间会大大增加，另一种常见的策略是分布计算：先计算出表达数据与所有协变量线性拟合的残差，然后用残差与基因型进行回归分析。

R Code：

1. gene10res <- resid(lm(express2$gene\_10 ~ as.matrix(covariates[,-1]))) *#残差*
2. gene10fit5<-lapply(genotype[,-1],function(x){lm(gene10res~x)}) *#表达残差与所有SNP简单线性回归*
3. pval4 <- sapply(gene10fit5,function(x){summary(x)$coef["x",]})[4,]
4. pval4

Return：

       snp\_1        snp\_2        snp\_3        snp\_4        snp\_5

1.273150e-01 7.300981e-01 7.980512e-02 6.557778e-01 5.730895e-01

       snp\_6        snp\_7        snp\_8        snp\_9       snp\_10

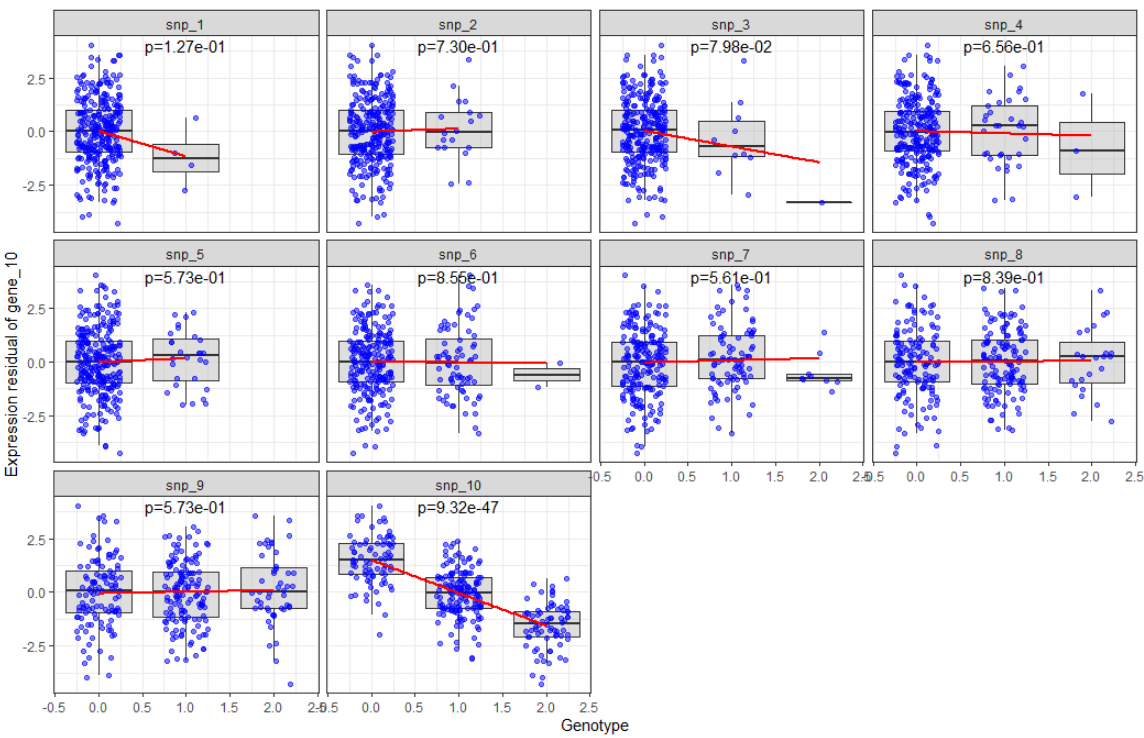
8.551790e-01 5.612916e-01 8.393056e-01 5.731244e-01 9.320975e-47

得到的结果与将所有协变量直接纳入分析模型的方法近似。我们可以用协变量修正后的残差数据绘制基因型与基因表达残差值的关系图。

R Code：

1. gene10\_snp3=cbind(genotype[,-1],gene10res)
2. plotdata7=data.table::melt(gene10\_snp3,id.vars="gene10res")
3. textdata3=data.frame(variable=names(pval4),label=formatC(pval4,digits=2,format="e"))
4. ggplot(data=plotdata7,aes(x=value,y=gene10res))+
5. geom\_boxplot(aes(group=value),outlier.shape=NA,fill="gray",alpha=0.5) +
6. geom\_jitter(color="blue",alpha = 0.5,position=position\_jitter(0.25))+
7. geom\_smooth(method="lm",se=F,color="red",formula=y~x) +
8. geom\_text(data=textdata3,aes(x=1,y=4,
9. label=paste0("p=",label)))+
10. facet\_wrap(~variable)+
11. labs(x = "Genotype",y = "Expression residual of gene\_10") +
12. theme\_bw()

Return：

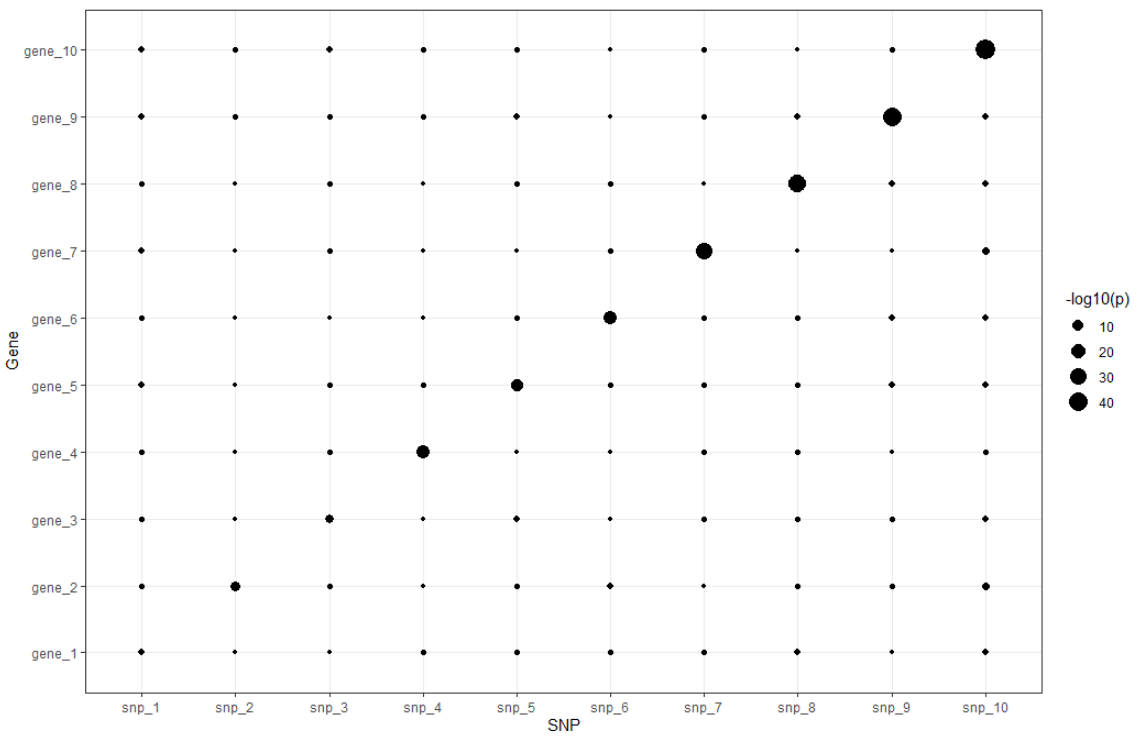


我们可以计算所有SNP和所有基因两两之间的关联性，根据得到的结果，绘制SNP和每个基因表达之间的关联图谱。

R Code：

1. generes <- sapply(express2[,-1],function(x){
2. resid(lm(x ~ as.matrix(covariates[,-1])))}) *#所有基因与所有协变量的残差值*
3. allcomb<-data.table::CJ(snp=colnames(genotype)[-1],gene=colnames(generes))
4. allpval=apply(allcomb,1,function(x){data.frame(snp=x[1],gene=x[2],p=coef(summary(lm(generes[,x[2]]~unlist(genotype[,x[1],with=F]))))[2,4])}) *#所有SNP和基因残差两两之间的显著性*
5. allpval<-do.call(rbind,allpval)
6. rownames(allpval)=NULL
7. allpval$snp=factor(allpval$snp,levels=colnames(genotype)[-1])
8. allpval$gene=factor(allpval$gene,levels=colnames(generes))
9. ggplot(allpval,aes(x=snp,y=gene))+
10. geom\_point(aes(size=-log10(p)))+
11. labs(x = "SNP",y = "Gene") +
12. theme\_bw()

Return：



结果展示了所有SNP和基因表达残差值关联的两两显著性结果，每个点的大小是p值的负对数，点越大，显著性越大。