作业

Homework1

庄镇华 502022370071

A Bioinformatics Homework Assignment

2022 年 11 月 28 日

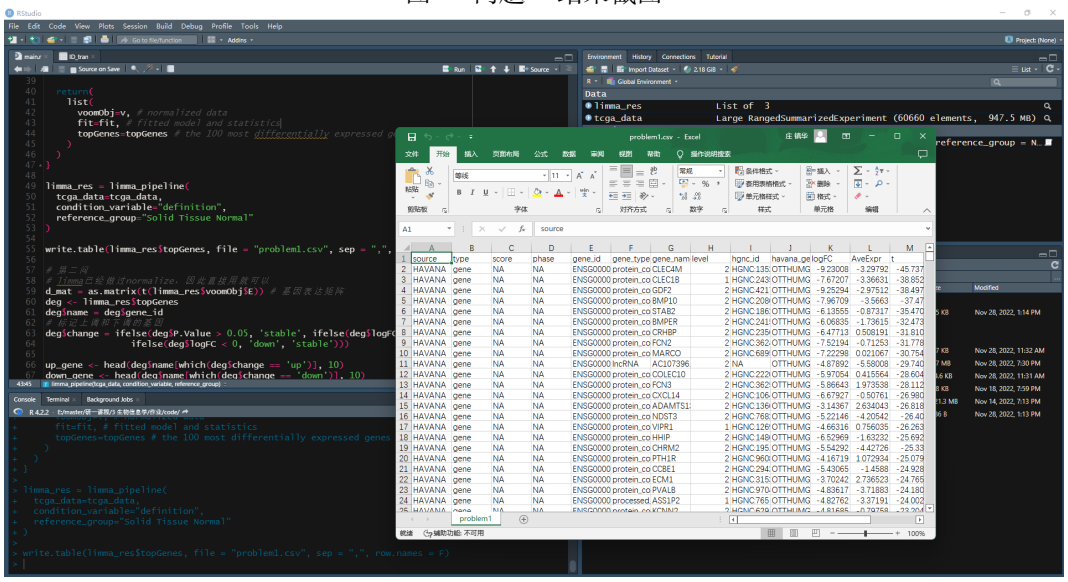
2022 年 11 月 28 日

问题 1: 使用 EdgeR 分析可以加 20% 分

使用 EdgeR 或 limma 对数据进行 Differential Expression 分析，对比 Tmuor samples 和 Normal samples，将结果保存为 problem1.csv。对于 DE 分析的数据集使用 tcga_data@colData$definition，即作业参考网站中 DE 分析所用的训练集部分即可。

解答:

图 1: 问题 1 结果截图



依据 RNA-Seq 数据可以进行基于某些临床表现型的 Differential Expression 分析，本次实验使用 limma 对数据进行 Differential 分析，由于批次不同以及其他因素的影响，需要在进行差异分析前将数据进行归一化处理。实现的主体流程在函数 limma_pipeline 里，下面对该函数进行详细解释。

函数的输入是 tcga_data 即原始数据以及用于对样本分组的条件变量，输出三个变量: 经过 voom 过程后的 TMM+voom 归一化数据、eBayes 拟合的一些模型以及每个探针相关的统计信息、依据 p-value 排序的差异表达基因。

```
limma_pipeline = function(tcga_data,
    condition_variable,
    reference_group=NULL){
        ...
  return(list(voomObj=v, # normalized data
      fit=fit, # fitted model and statistics
      topGenes=topGenes # the 100 most differentially expressed genes
    ))
}
```

首先，我们得到依据肿瘤和正常样本分组的样本，并定义正常组织作为参照类别。

```
design_factor = colData(tcga_data)[, condition_variable, drop=T]
group = factor(design_factor)
if(!is.null(reference_group)){group = relevel(group, ref=reference_group)}
```

然后，我们创建一个矩阵用于指示 DE 分析所要比较的条件，接着我们删掉计数过少的基因，并且将 tcga_data 对象转化为 DGEList 格式方便后续处理。

```
dge = DGEList(counts=assay(tcga_data),
      samples=colData(tcga_data),
      genes=as.data.frame(rowData(tcga_data)))
# filtering
keep = filterByExpr(dge,design)
dge = dge[keep,,keep.lib.sizes=FALSE]
rm(keep)
```

接着，我们用 TMM 归一化方法对数据进行归一化处理。

```
# Normalization (TMM followed by voom)
dge = calcNormFactors(dge)
v = voom(dge, design, plot=TRUE)
```

最后，使用 lmFit 生成拟合数据的线性模型，用 eBayes 处理这些线性模型生成用于最终排序的统计量，用 topTable 函数对差异化表达基因排序，这里的 number=Inf 代表保留所有结果。

```
# Fit model to data given design
fit = lmFit(v, design)
fit = eBayes(fit)
# Show top genes
topGenes = topTable(fit, coef=ncol(design), number=Inf, sort.by="p")
```

最终的结果保存在 problem1.csv 文件里，表1展示结果的前 5 行。

表 1: Differential Expression 分析部分结果

source	type	score	phase	gene_id	gene_type	gene_name
HAVANA	gene	NA	NA	ENSG00000104938.18	protein_coding	CLEC4M
HAVANA	gene	NA	NA	ENSG00000165682.14	protein_coding	CLEC1B
HAVANA	gene	NA	NA	ENSG00000263761.3	protein_coding	GDF2
HAVANA	gene	NA	NA	ENSG00000163217.2	protein_coding	BMP10
HAVANA	gene	NA	NA	ENSG00000136011.15	protein_coding	STAB2

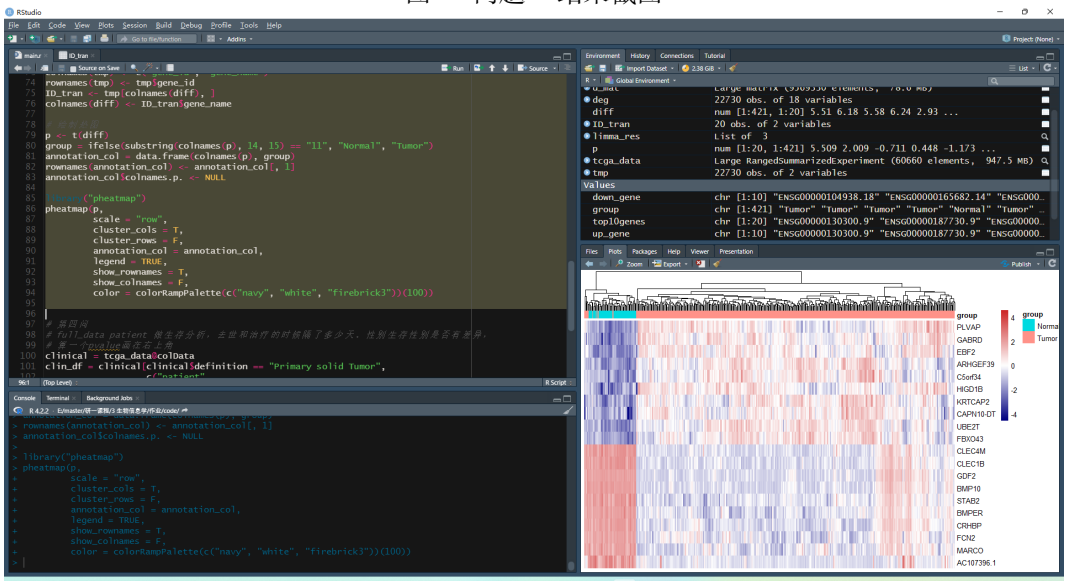
hgnc_id	havana_gene	logFC	t	P.Value	adj.P.Val
HGNC:13523	OTTHUMG00000182432.4	-9.23008	-45.7372	7.19e-166	1.63E-161
HGNC:24356	OTTHUMG00000168502.1	-7.67207	-38.8524	1.18e-141	1.34E-137
HGNC:4217	OTTHUMG00000188320.2	-9.25294	-38.4971	2.45e-140	1.86E-136
HGNC:20869	OTTHUMG00000129573.2	-7.96709	-37.472	1.69e-136	9.59E-133
HGNC:18629	OTTHUMG00000170056.2	-6.13555	-35.4704	7.67e-129	3.49E-125

问题 2: 50%

对问题 1 中 Differential Expression 的结果，分别选出 TOP10 significant Tumor 基因 (Tumor 高于 Normal),、TOP10 significant Normal 基因 (Normal 高于 Tumor)，并使用 pheatmap 将以上结果画成热图。

解答:

图 2: 问题 2 结果截图



问题 2 的具体流程如下所述，最终生成的 Top10 基因如表2所示，热图如图3所示。需要注意在使用 pheatmap 包绘制热图时也需要进行归一化处理，由于已经保存了 limma 归一化处理的结果，因此直接使用即可。

利用 pvalue >0.05 选出显著表达的基因，标记上调和下调的基因。

```
# limma已经做过normalize，因此直接用就可以
d_mat = as.matrix(t(limma_res$voomObj$E)) # 基因表达矩阵
deg <- limma_res$topGenes
deg$name = deg$gene_id
# 标记上调和下调的基因
deg$change = ifelse(deg$P.Value > 0.05, 'stable', ifelse(deg$logFC > 0, 'up',
                 ifelse(deg$logFC < 0, 'down', 'stable')))
```

选出前 10 个 signigicant Tumor 基因，前 10 个 signigicant Normal 基因，最后利用 pheatmap 绘制热图。

```
up_gene <- head(deg$name[which(deg$change == 'up')], 10)
down_gene <- head(deg$name[which(deg$change == 'down')], 10)
top10genes <- c(as.character(up_gene), as.character(down_gene))
diff = d_mat[, top10genes]

# 利用gene_id获取gene_name
tmp <- data.frame(deg$gene_id, deg$gene_name)
colnames(tmp) <- c("gene_id", "gene_name")
rownames(tmp) <- tmp$gene_id
ID_tran <- tmp[colnames(diff), ]
colnames(diff) <- ID_tran$gene_name

# 绘制热图
p <- t(diff)
group = ifelse(substring(colnames(p), 14, 15) == "11", "Normal", "Tumor")
annotation_col = data.frame(colnames(p), group)
rownames(annotation_col) <- annotation_col[, 1]
```

```r
annotation_col$colnames.p. <- NULL

library("pheatmap")
pheatmap(p,
         scale = "row",
         cluster_cols = T,
         cluster_rows = F,
         annotation_col = annotation_col,
         legend = TRUE,
         show_rownames = T,
         show_colnames = F,
         color = colorRampPalette(c("navy", "white", "firebrick3"))(100))
```

图 3: Top10 significant 基因热图

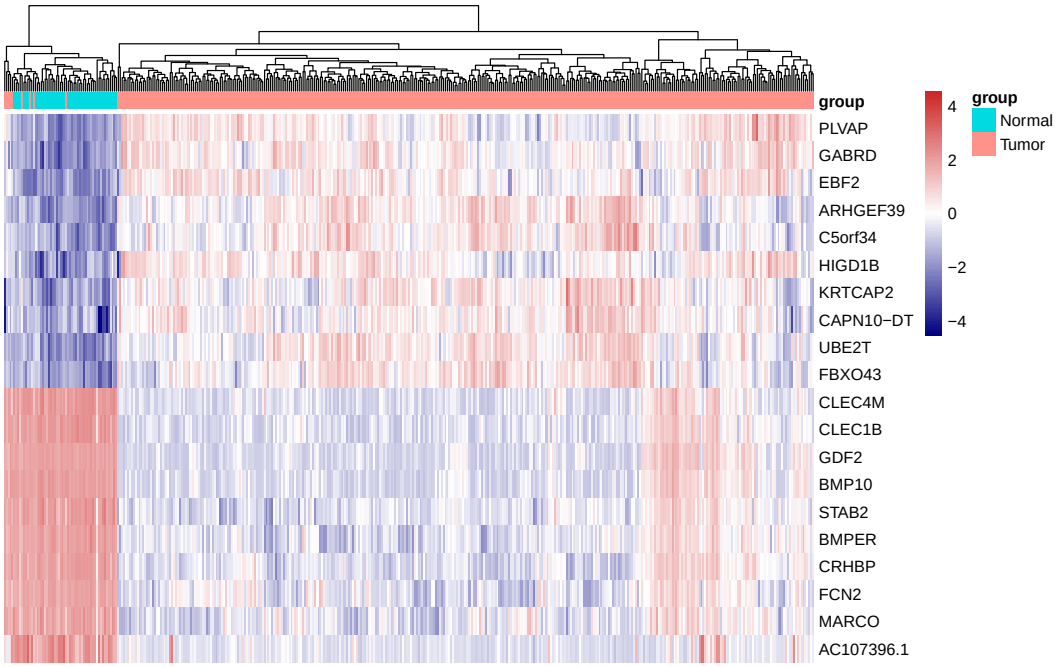


表 2: Top10 significant 基因

Top10 tumor gene	PLVAP	GABRD	EBF2	ARHGEF39	C5orf34
	HIGD1B	KRTCAP2	CAPN10-DT	UBE2T	FBXO43
Top10 normal gene	CLEC4M	CLEC1B	GDF2	BMP10	STAB2
	BMPER	CRHBP	FCN2	MARCO	AC107396.1

问题 3: 附加题 20%

对问题 1 中 Differential Expression 的结果分别根据 pvalue 和 logFC 排序得到两个 gene ranked list，使用 GSEA 软件分别对两个 gene ranked list 进行分析。

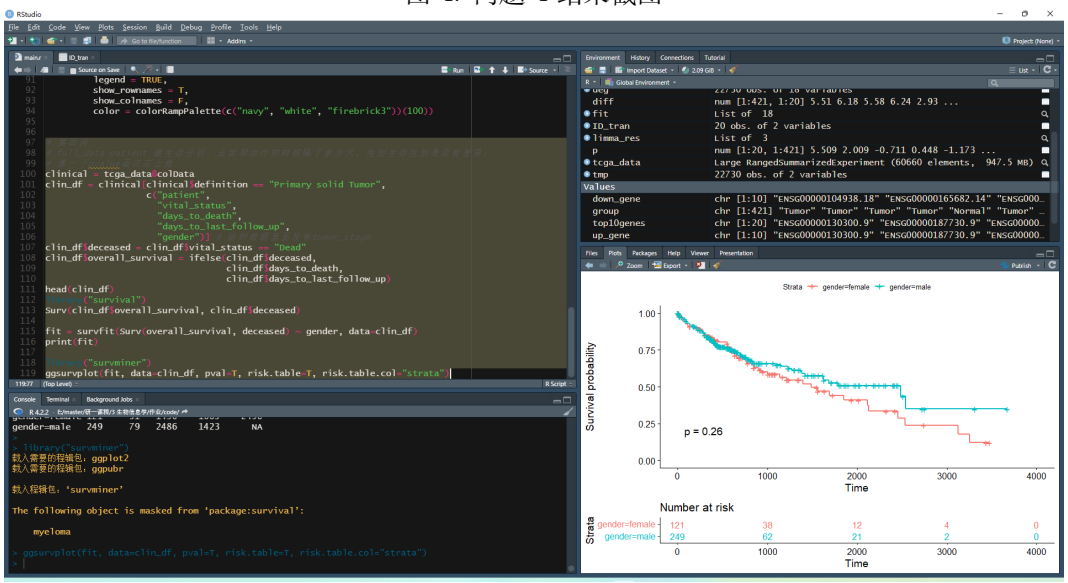
解答:

问题 4: 附加题 20%

使用 Survival 包进行关于 Gender 的 survival analysis。

解答:

图 4: 问题 4 结果截图



利用 survial 包对性别变量进行生存分析，即分析是否某一组患者比另一组患者可能活得更长。具体流程如下：

提取性别变量以及其他相关变量，例如：是否生存、肿瘤阶段、初诊距离死亡时间等。

```
clinical = tcga_data@colData
clin_df = clinical[clinical$definition == "Primary solid Tumor",
                   c("patient",
                     "vital_status",
                     "days_to_death",
                     "days_to_last_follow_up",
                     "gender")] # 给的数据里面没有tumor_stage
clin_df$deceased = clin_df$vital_status == "Dead"
clin_df$overall_survival = ifelse(clin_df$deceased,
                                  clin_df$days_to_death,
                                  clin_df$days_to_last_follow_up)
```

创建生存公式并将之传给 survfit 函数生成 Kaplan-Meier 图。

```
library("survival")
Surv(clin_df$overall_survival, clin_df$deceased)
fit = survfit(Surv(overall_survival, deceased) ~ gender, data=clin_df)

library("survminer")
ggsurvplot(fit, data=clin_df, pval=T, risk.table=T, risk.table.col="strata")
```

分析结果如下，可以看到在时间超过 2000 天之后，女性患者的生存概率小于男性，但由于超过 2000 天的样本量过少，因此 p-value 为 0.26 >> 0.05，结果并不显著，可以认为性别对生存时间基本没有显著影响。

图 5: 关于 gender 的生存分析图