This article provides some picture code, and the remaining pictures are generated by the websites of biological informatics, which are clearly described in the article.

Fig 1 A, B

（一）

# 安装和加载**ggplot2**包

install.packages("ggplot2")

library(ggplot2)

# 假设你已经从TCGA数据库获取了基因表达数据，将其存储在一个数据框中

# 示例数据框，需要替换为你的数据

gene\_expression\_data <- data.frame(

Gene = c("Gene SEMA4D"), # 基因名

Cancer\_Type = c("Breast Cancer", "Lung Cancer", "Colon Cancer"), # 癌症类型

Expression = c(2.3, 3.5, 1.8) # 基因表达值

)

# 使用ggplot2创建泛癌基因表达可视图

ggplot(gene\_expression\_data, aes(x = Cancer\_Type, y = Expression, fill = Gene)) +

geom\_bar(stat = "identity", position = "dodge") +

labs(title = "TCGA泛癌基因表达可视图", x = "癌症类型", y = "表达值") +

theme\_minimal()

（二）

# Load the stats package

library(stats)

# Load the TCGA data

tcga\_data <- read.csv("TCGA\_data.csv")

# Get the expression values for the gene of interest

gene\_expression <- tcga\_data$gene\_expression

# Create a histogram of the expression values

hist(gene\_expression)

# Calculate the mean and standard deviation of the expression values

mean\_expression <- mean(gene\_expression)

sd\_expression <- sd(gene\_expression)

# Print the mean and standard deviation of the expression values

print(mean\_expression)

print(sd\_expression)

（三）

# Load the car package

library(car)

# Load the TCGA data

tcga\_data <- read.csv("TCGA\_data.csv")

# Get the expression values for the gene of interest

gene\_expression <- tcga\_data$gene\_expression

# Create a histogram of the expression values

hist(gene\_expression)

# Calculate the mean and standard deviation of the expression values

mean\_expression <- mean(gene\_expression)

sd\_expression <- sd(gene\_expression)

# Print the mean and standard deviation of the expression values

print(mean\_expression)

print(sd\_expression)

```

Fig 2 B-D

（四）

library(ggalluvial)

df <- data.frame(

start = c("a", "a", "b", "b", "c"),

end = c("a", "b", "b", "c", "c"),

count = c(3, 2, 3, 1, 5))

ggalluvial(

data = df,

x = "start",

y = "count",

z = "end")

Fig 5

(五)

library(pROC)

true <- c(1, 1, 1, 0, 0, 0)

pred <- c(0.9, 0.8, 0.4, 0.3, 0.2, 0.1)

roc\_obj <- roc(response = true, predictor = pred)

plot(roc\_obj, print.threshold = TRUE, text.thres="Cutoff Point",

main="ROC Curve", xlab="False Positive Rate", ylab="True Positive Rate")

auc\_value <- auc(roc\_obj)

text(x=0.5, y=0.5, paste("AUC=", round(auc\_value, 2)))

auc\_value

Fig 10

**ggplot2**包同（一）