BIOF3 组学数据分析

07 机器学习预测模型

导出日期：2026年5月12日

07 机器学习预测模型

前面几个模块侧重"发现结构"（因子、模块、亚型），本模块转向"预测"：用多组学特征预测临床结局（如药物响应、生存状态、疾病亚型）。核心问题是如何从多层高维数据中选出有预测力的特征子集，以及如何设计合理的交叉验证策略。

特征选择策略

多组学数据的特征数远大于样本数（p >> n），直接建模会过拟合。常见的特征选择路径：

1. **单层筛选后拼接**：每层独立做方差过滤或差异分析，保留 top 特征，再拼接
2. **用整合结果做特征**：MOFA 因子得分、WGCNA 模块 eigengene 作为输入
3. **嵌入式选择**：用 elastic net 或 Random Forest 的内置特征重要性

```
# 方法 1：每层取 top 500 高变异特征
select_top <- function(mat, n = 500) {
  rv <- apply(mat, 1, var)
  mat[names(sort(rv, decreasing = TRUE))[1:n], ]
}

rna_sel  <- select_top(rna_final, 500)
meth_sel <- select_top(meth_final, 500)
prot_sel <- select_top(prot_final, 200)

# 拼接
X <- t(rbind(rna_sel, meth_sel, prot_sel))
y <- factor(col_data$subtype)
```

Elastic Net 分类

Elastic net 结合了 L1（特征选择）和 L2（稳定性）正则化，适合高维小样本场景：

```r
library(glmnet)

# alpha = 0.5 是 elastic net；alpha = 1 是 lasso
set.seed(42)
cv_fit <- cv.glmnet(X, y, family = "multinomial",
                    alpha = 0.5, nfolds = 5)

# 最优 lambda
plot(cv_fit)
best_lambda <- cv_fit$lambda.min

# 非零系数（被选中的特征）
coef_list <- coef(cv_fit, s = best_lambda)
selected_features <- lapply(coef_list, function(c) {
  rownames(c)[which(c[, 1] != 0)]
})
```

Random Forest

Random Forest 不需要特征预筛选，能处理非线性关系，并提供特征重要性排序：

```r
library(randomForest)

set.seed(42)
rf_model <- randomForest(X, y, ntree = 1000, importance = TRUE)

# OOB 错误率
print(rf_model)

# 特征重要性
imp <- importance(rf_model, type = 1)
top_imp <- head(sort(imp[, 1], decreasing = TRUE), 30)
barplot(top_imp, las = 2, main = "Top 30 features by importance",
        cex.names = 0.6)
```

交叉验证设计

多组学预测模型最容易犯的错误是信息泄露（data leakage）。特征选择必须在交叉验证的内层完成，不能用全部数据选完特征再做 CV。

```
library(caret)

# 正确做法：用 caret 的嵌套 CV
ctrl <- trainControl(method = "repeatedcv",
                     number = 5,
                     repeats = 10,
                     classProbs = TRUE,
                     summaryFunction = multiClassSummary)

# 在每个 fold 内部做特征选择 + 建模
set.seed(42)
rf_cv <- train(X, y,
               method = "rf",
               trControl = ctrl,
               tuneLength = 5,
               metric = "AUC")
print(rf_cv)
```

如果样本量很小（< 50），考虑 leave-one-out CV（LOOCV）或 repeated 5-fold CV 来减少方差。

多组学 vs 单组学的预测力比较

一个关键问题是：多组学整合是否真的比单组学预测更好？需要做对照实验：

```
# 分别用单层数据训练
rf_rna  <- train(t(rna_sel), y, method = "rf", trControl = ctrl, metric = "AUC")
rf_meth <- train(t(meth_sel), y, method = "rf", trControl = ctrl, metric = "AUC")
rf_prot <- train(t(prot_sel), y, method = "rf", trControl = ctrl, metric = "AUC")

# 比较
results <- resamples(list(
  MultiOmics = rf_cv,
  RNA_only   = rf_rna,
  Meth_only  = rf_meth,
  Prot_only  = rf_prot
))
summary(results)
dotplot(results, metric = "AUC")
```

如果多组学没有显著提升，说明信息冗余或者某一层已经足够。这本身也是有价值的结论。

生存预测

如果目标是生存时间而不是分类，用 Cox 回归配合 elastic net：

```
library(glmnet)

surv_y <- Surv(col_data$os_time, col_data$os_event)

cv_cox <- cv.glmnet(X, surv_y, family = "cox",
                    alpha = 0.5, nfolds = 5)
# C-index
max(cv_cox$cvm)
```

参考资源

- glmnet vignette
- caret 包文档
- randomForest 包
- 交叉验证中的信息泄露 (Hastie et al. ESL Ch.7)
- 多组学预测综述 (Picard et al. 2021)

扫码关注微信公众号【生信F3】

获取文章完整内容，分享生物信息学最新知识。