BIOF3 组学数据分析

02 各组学数据清洗与特征对应

导出日期：2026年5月12日

02 各组学数据清洗与特征对应

每一层组学数据在进入整合分析之前，都需要独立完成预处理。整合方法不会帮你处理批次效应、低质量探针或未归一化的计数值——垃圾进去，垃圾出来。

各层预处理要点

转录组（RNA-seq）

```
library(DESeq2)

dds <- DESeqDataSetFromMatrix(countData = rna_counts,
                              colData   = sample_info,
                              design    = ~ 1)
# 过滤低表达基因
keep <- rowSums(counts(dds) >= 10) >= 5
dds <- dds[keep, ]

# 方差稳定化（用于下游整合，不是差异分析）
vsd <- vst(dds, blind = TRUE)
rna_mat <- assay(vsd)
```

对于整合分析，通常用 VST 或 rlog 变换后的矩阵，而不是原始 counts 或 TPM。

甲基化（450K / EPIC）

```
library(minfi)

# 从 IDAT 读入
rgSet <- read.metharray.exp(targets = targets)
# 预处理：Noob 背景校正 + dye-bias 校正
mSet <- preprocessNoob(rgSet)
# 获取 beta 值
beta <- getBeta(mSet)

# 过滤：去掉 SNP 探针、cross-reactive 探针、性染色体探针
beta <- beta[!rownames(beta) %in% snp_probes, ]
beta <- beta[!rownames(beta) %in% cross_reactive, ]
beta <- beta[!grepl("^ch\\.", rownames(beta)), ]
```

蛋白质组

蛋白质组数据通常已经是 log2 intensity。主要处理缺失值和批次效应：

```r
# 过滤缺失率 > 50% 的蛋白
miss_rate <- rowMeans(is.na(prot_mat))
prot_mat <- prot_mat[miss_rate < 0.5, ]

# KNN 填补
library(impute)
prot_mat <- impute.knn(prot_mat)$data

# 批次校正
library(limma)
prot_mat <- removeBatchEffect(prot_mat, batch = batch_info)
```

特征对应：从探针到基因

整合分析需要不同组学的特征能对应到同一个实体（通常是基因）。甲基化探针和蛋白 ID 都需要映射到 gene symbol。

甲基化探针 → 基因

```r
library(IlluminaHumanMethylation450kanno.ilmn12.hg19)

anno <- getAnnotation(IlluminaHumanMethylation450kanno.ilmn12.hg19)
probe_gene <- data.frame(
  probe = anno$Name,
  gene  = anno$UCSC_RefGene_Name,
  stringsAsFactors = FALSE
)
# 一个探针可能对应多个基因，取第一个或做 promoter 过滤
probe_gene$gene <- sapply(strsplit(probe_gene$gene, ";"), `[`, 1)
probe_gene <- probe_gene[probe_gene$gene != "", ]
```

基因级别聚合

一个基因可能对应多个甲基化探针。常见做法是取 promoter 区域探针的均值：

```r
# 只保留 TSS200 和 TSS1500 区域的探针
promoter_probes <- anno$Name[grepl("TSS", anno$UCSC_RefGene_Group)]

beta_promoter <- beta[rownames(beta) %in% promoter_probes, ]

# 按基因聚合（取均值）
probe_gene_sub <- probe_gene[probe_gene$probe %in% rownames(beta_promoter), ]
beta_gene <- aggregate(beta_promoter[probe_gene_sub$probe, ],
                       by = list(gene = probe_gene_sub$gene),
                       FUN = mean)
rownames(beta_gene) <- beta_gene$gene
beta_gene$gene <- NULL
```

蛋白 → 基因

```
library(biomaRt)

ensembl <- useMart("ensembl", dataset = "hsapiens_gene_ensembl")
prot_ids <- rownames(prot_mat)  # UniProt ID

mapping <- getBM(attributes = c("uniprotswissprot", "hgnc_symbol"),
                 filters    = "uniprotswissprot",
                 values     = prot_ids,
                 mart       = ensembl)
# 合并
prot_mat_gene <- prot_mat[mapping$uniprotswissprot, ]
rownames(prot_mat_gene) <- mapping$hgnc_symbol
```

对齐后的检查

预处理和映射完成后，确认三层数据的维度和样本一致：

```
# 确保列（样本）顺序一致
common_samples <- intersect(intersect(colnames(rna_mat), colnames(beta_gene)),
                            colnames(prot_mat_gene))
rna_final  <- rna_mat[, common_samples]
meth_final <- beta_gene[, common_samples]
prot_final <- prot_mat_gene[, common_samples]

cat("样本数:", length(common_samples), "\n")
cat("RNA 特征:", nrow(rna_final), "\n")
cat("甲基化特征:", nrow(meth_final), "\n")
cat("蛋白特征:", nrow(prot_final), "\n")
```

参考资源

- DESeq2 vignette
- minfi 预处理流程
- biomaRt 使用指南
- impute 包