RCTD多切片分析：复用参考对象，逐片运行并检查权重

固定spacexr接口和参考数据，把每张切片作为独立任务运行；检查基因、坐标、模式和输出权重，避免把接口差异或资源试跑误当成生物学结论。

RCTD用带细胞类型注释的单细胞参考，分解空间转录组每个spot的细胞类型组成。多切片时可以复用同一份Reference，但每张切片仍需核对counts、坐标和spot名称，再独立建立空间对象与RCTD模型。本例固定为dmcable/spacexr仓库的Reference、SpatialRNA、create.RCTD、run.RCTD接口；Bioconductor的spacexr使用createRctd/runRctd及SpatialExperiment，二者不要混写。

先核对数据契约

对象	必要内容	重点检查
reference.rds	counts、细胞类型、nUMI	原始计数；列名与标签对应
每片输入RDS	counts、coords、nUMI	行列名称、坐标spot一致
slices.csv	slice、input两列	切片名唯一，路径存在

spacexr对象要求基因为矩阵行、细胞或spot为列；SpatialRNA的coords以spot为行，包含数值x、y坐标，counts使用未变换的计数。Reference也要使用未归一化计数，细胞类型标签必须按细胞名对应。建对象前取参考与空间数据共有基因，并按同一顺序重排；不要只比较交集数量而忽略行名次序。每张切片的spot条码应唯一，坐标行名必须与counts列名完全对应。

复用参考、逐片保存结果

先在固定环境中生成Reference并保存，所有切片共用该对象。下面假定每个切片RDS是列表，字段为counts、coords、nUMI；按实际导出格式改字段映射。为每片建立独立目录，保存模型、结果、参数和日志。先用一张代表性切片及较小max_cores验证输入/流程，再按切片依次运行，避免多个重任务同时争抢内存。

```
library(spacexr)
stopifnot(as.character(packageVersion("spacexr")) == "2.2.1")
ref <- readRDS("reference.rds")
stopifnot(inherits(ref, "Reference"))
jobs <- read.csv("slices.csv", stringsAsFactors=FALSE)
stopifnot(all(c("slice","input") %in% names(jobs)),
          nrow(jobs) > 0, !anyNA(jobs),
          !anyDuplicated(jobs$slice), all(file.exists(jobs$input)))
dir.create("rctd-results", showWarnings=FALSE)
for (i in seq_len(nrow(jobs))) {
  x <- readRDS(jobs$input[i]) # list: counts, coords, nUMI
  counts <- x$counts
  coords <- x$coords
  stopifnot(!is.null(rownames(counts)), !is.null(colnames(counts)),
            !anyDuplicated(rownames(counts)), !anyDuplicated(colnames(counts)),
            all(c("x","y") %in% colnames(coords)),
            setequal(colnames(counts), rownames(coords)),
            !is.null(x$nUMI), all(colnames(counts) %in% names(x$nUMI)))
  coords <- coords[colnames(counts), c("x","y"), drop=FALSE]
  common <- intersect(rownames(ref@counts), rownames(counts))
  stopifnot(length(common) > 0)
  counts <- counts[common, , drop=FALSE]
  r <- Reference(ref@counts[common, , drop=FALSE],
                 ref@cell_types, ref@nUMI)
  space <- SpatialRNA(coords, counts, x$nUMI[colnames(counts)])
  model <- create.RCTD(space, r, max_cores=4)
  model <- run.RCTD(model, doublet_mode="full")
  saveRDS(model, file.path("rctd-results", paste0(jobs$slice[i], ".rds")))
}
```

代码展示逐片运行结构，不替代真实输入验证。Reference若含需同步处理的其他槽或附加元数据，应按固定版本构造器重新生成，而不是只改counts槽。mode需按测量分辨率选择：doublet限制每个spot最多两类，multi限制类别数，full不施加这一限制。示例用full仅表示无类别数上限，不代表适用于所有技术；按照平台空间单位和研究问题选择，并记录理由。

资源试跑和正式分析分开

RCTD主要使用CPU并行；先从单片、max_cores=1或较小值开始，记录输入spot数、共有基因数、参考细胞数、耗时、主机峰值内存和日志，再逐步调整核心数。核数增加不一定线性加速，也可能增加内存竞争。参考对象只读复用；不要在同一目录并行写同名模型文件。

试跑只验证对象构造、模式和资源范围。正式结果检查每片运行日志、spot覆盖率、权重范围及空间分布；若选doublet模式，再统计spot_class中的reject与uncertain比例。full模式权重按该版本文档归一化后解释。逐片独立拟合不会自动校正切片间技术差异，跨片比较前需确认模式、参数和输入流程一致。Bioconductor版的权重矩阵方向与GitHub版不同，不能直接套用索引或可视化代码。若某类细胞铺满组织或大量spot被拒绝，先检查参考是否覆盖组织细胞、共有基因和标签，再判断模型参数，不能单靠加核解决。

保存可复核记录

记录软件来源和版本、参考文件哈希、每片输入哈希、模式、共有基因列表、max_cores、开始结束时间与输出文件。按切片分别保存结果，确认每个结果中的spot条码能回连原始坐标，再做跨切片比较。只有代表性试跑显示当前设备无法满足项目时限时，才根据切片数、spot数和并发需求评估额外计算资源。

参考资料

dmcable/spacexr官方数据对象与RCTD接口说明 官方文档
Bioconductor版spacexr RCTD教程及结果结构 官方文档
spacexr官方2.2.1手册 官方文档