SCENIC+部署与试跑教程：锁定依赖、数据库和阶段资源

完整操作教程 · 参考固定 · 往返恢复验收

固定SCENIC+源码与Python范围，按生成配置核对多组学对象、数据库及Snakemake阶段。

运行边界：示例代码需按安装版本和真实项目输入复核；未提供真实生物数据，语法或小样本检查不代表科学结果和性能验收。

SCENIC+部署失败常来自依赖、对象和数据库不匹配，未必需要更多CPU或内存。应先固定源码修订和Python范围，跑通命令入口，再填写工作流配置并做dry-run。本篇以当前核对源码及Snakemake入口说明，不把导入成功当作多组学网络已经可信。

先锁定代码与依赖范围

本次核对的main修订为840dab85de3044846234c157e327b6ea0290abfb，项目元数据标1.0a2且Python要求不高于3.11.8；网页教程仍显示1.0a1。二者不能按“latest”混用。以下在已安装Conda与Git的Linux新目录构建独立环境，记录实际依赖解算结果。

```
#!/usr/bin/env bash
set -euo pipefail
mkdir scenicplus-deploy-001
cd scenicplus-deploy-001
envprefix="$PWD/scplus-env-001"
git clone https://github.com/aertslab/scenicplus.git scenicplus-source
git -C scenicplus-source checkout 840dab85de3044846234c157e327b6ea0290abfb
git -C scenicplus-source rev-parse HEAD > source-revision.txt
conda create -y -p "$envprefix" python=3.11.8 pip
conda run -p "$envprefix" python -m pip install ./scenicplus-source
conda run -p "$envprefix" python -m pip check
conda run -p "$envprefix" python -c 'import scenicplus; print(scenicplus.__version__)'
conda run -p "$envprefix" scenicplus --help > cli-help.txt
conda list -p "$envprefix" --explicit > conda-explicit.txt
conda run -p "$envprefix" python -m pip freeze > pip-freeze.txt
mkdir scplus-run-001
conda run -p "$envprefix" scenicplus init_snakemake --out_dir "$PWD/scplus-run-001"
```

pip check主要核包依赖元数据，不会验证数据库、所有本地二进制或真实分析。遇到版本约束先核该修订要求和完整错误，不在旧环境反复强制升级numpy等底层包。若需要补其他工作流依赖，按同一修订的安装材料处理，并再次保存环境清单；不要凭软件名称自动申请GPU。

按生成模板填写数据与数据库

阶段	固定材料	验收内容
细胞输入	cisTopic_obj_fname、GEX_anndata_fname	对象可读、条码与样本映射正确
区域集合	region_set_folder	共识峰、坐标与分组结构
motif数据库	ctx_db_fname、dem_db_fname	ranking/score类型、组装与区域范围
注释	path_to_motif_annotations及物种配置	motif版本、TF命名与基因ID
输出与资源	output_data、临时目录和工作流线程	新路径、剩余工作盘与获配额度

编辑生成的Snakemake/config/config.yaml，以安装版模板为准，不整段覆盖成其他版本配置。配对multiome需验收RNA与ATAC的细胞对应；非配对数据需要适当的分组与metacell设计，不能只截取条码相同部分冒充配对。RNA表达表示和raw的语义也应按该入口确认。

先dry-run，再做代表任务

数据库路径、输出目录、物种与方法参数填好后，在同一环境执行以下命令。第二段在前面相同Bash会话继续，当前目录仍为scenicplus-deploy-001，envprefix仍为之前的绝对路径；若换终端，先回到部署目录并重新设置这个变量。dry-run主要检查DAG和输入依赖，不读取并验证所有对象内容；它失败时先修配置，不立即提高资源。实际小任务也应包含输入准备、motif富集和网络步骤，而非只执行help。

```
set -euo pipefail
cd scplus-run-001/Snakemake
conda run -p "$envprefix" snakemake --dry-run --cores 4 > dry-run.log 2>&1
/usr/bin/time -v -o resources.txt \
  conda run -p "$envprefix" snakemake --cores 4 > workflow.log 2>&1
```

此处4核是本地工作流的演示总额度，仍需检查规则内部线程与其他数值库。记录每阶段RAM、墙钟、数据库载入和临时盘，不能只看最后一步。对象或数据库更新后先重审缓存依赖，不把存在的中间文件一律当作与新输入兼容。失败现场保留，重跑前明确哪些输出已通过验收。

把环境错误与研究验收分开

下载数据库前先核物种、组装、区域覆盖及motif注释是否成套，分别记录来源和文件校验值。磁盘规划包括数据库原件、中间对象、临时文件和旧运行结果共存；下载成功不代表数据库内容已与本项目匹配。

ImportError、ABI或命令缺失先定位软件栈；输入基因和区域覆盖异常先查数据与数据库；作业Killed再查实际内存与平台记录。仅换大机器不能修正错误物种、组装或motif注释，也不能把不合适的主题/区域集合自动变成可靠调控网络。

最终验收包括细胞对应、候选区域、motif富集、TF—区域—基因关系及必要的独立证据。网络推断不直接证明因果调控。保存源码SHA、环境清单、配置、数据库来源与校验值，使后续能解释版本差异；只有必要流程确有持续资源缺口时，才据实测CPU、RAM和工作盘规划环境。

先理解SCENIC+输入与工作流阶段

SCENIC+结合RNA表达、ATAC可及性、候选增强子和转录因子基序信息，通常由Snakemake管理多个依赖任务。开始前列出RNA细胞、ATAC细胞、peak集、TF/motif资源和基因组版本，确认细胞对应关系及样本过滤一致。输入对象通过格式检查不等于多组学细胞匹配和生物学假设已成立。

先用官方小数据或受控小样本跑通环境与规则文件，再执行真实项目。将工作流每一阶段的命令、输入、输出、日志、CPU/内存和运行时收集到独立目录。Snakemake规则的单步成功不等于整条流程完成；最终检查所有必需模块和最终regulon对象。

数据库准备和下载策略

cisTarget数据库和motif ranking资源可能体积较大，并且需与物种、基因组及数据库格式匹配。记录官方下载页面、release版本、压缩包散列、解压目录和安装日期；不要在工作流运行中重复下载或使用latest覆盖已验收数据库。若离线运行，先在联网环境按许可证和机构规范准备资源并验证完整性。

数据库所在磁盘需同时考虑压缩文件、解压文件、索引和工作流中间结果。读写权限与并发访问也需确认。若多个项目共享数据库，设为只读版本目录，并为每次流程建立自己的工作目录。数据库缺少或版本不匹配时，增加内存不能修复输入问题。

依赖、环境和工作流复现

SCENIC+涉及Python/R包、外部工具和Snakemake规则；仅有pip freeze不一定能固定系统库、命令行工具和数据库。保存conda/mamba环境锁文件、容器镜像摘要、SCENIC+提交版本、Snakemake版本、工作流参数和外部资源散列。环境升级后先做最小样本回归，确认规则仍能启动且关键输出结构一致。

工作目录中避免把不同版本生成的中间文件混在一起。Snakemake可能依据时间戳和规则输入判断是否复用结果；输入、规则或参数发生变化时，确认相应任务确实重跑。修改规则后记录diff和新的代码散列，不要只清除部分结果让依赖关系变得不透明。

资源配置与故障定位

每个规则的资源开销不同。按官方工作流的资源配置方式为规则设定线程、内存和运行时限，并从代表性数据测量峰值。不能把总线程数等同于整条流水线的并发额度，也不能只按细胞数估计数据库、motif扫描和网络推断阶段的内存。

某个rule失败时查看对应日志、输入文件完整性、系统OOM和Snakemake作业状态。磁盘满、参考数据库缺失、依赖导入失败、内存超限应分别处理。恢复时先确认已完成任务的输入和输出未被改变，再利用工作流依赖重跑；不要手工拼接不同运行目录中的中间文件。

regulon结果怎么解释

SCENIC+输出的候选enhancer-driven gene regulatory network依赖输入细胞、候选区域、motif匹配和统计阈值。网络边是模型推断，不是已验证因果调控。对重要TF-增强子-基因关系，结合TF表达、peak可及性、基因表达、已知调控数据或扰动实验验证。

如果不同运行版本或数据库得到的网络不同，比较输入峰、motif集合、数据库release和阈值，而不是只比较最终边数。报告候选网络覆盖、阈值、未匹配TF和被过滤区域，并保留中间证据链。

资源申请与交付

资源评估表写清细胞数、peak数、数据库体积、总任务数、并发数、各规则实际峰值RSS、临时盘和最长任务。正式项目以最慢或内存最大的阶段作为测试样本，不用一个教程小数据的时间外推全队列。

交付环境锁文件、版本清单、数据库manifest、Snakemake配置、输入散列、日志、最终对象和每阶段状态。注明哪些步骤实际跑过、哪些被跳过及原因。合规数据不要放入公开代码包，模板只含占位符或合成材料。

检查Snakemake状态和日志

运行前使用dry-run查看将执行的规则、预计输入和输出路径，确认不会读取其他项目或旧版本目录。指定合理的cores和资源配置；若用profile提交集群，应保存profile本身及调度器资源设置。完成后查看Snakemake汇总和每条失败规则日志，不能只看最终命令的退出码。

工作流自动重用中间结果依赖文件时间与规则关系。更换数据库、峰集或参数后，核实受影响规则确实重跑。可将关键输入散列加入项目manifest，帮助判断缓存是否有效。某些中间文件可再生，但不意味着可从无关运行目录拷贝并合并。

多组学对应与样本设计

若RNA与ATAC来自配对细胞或同一样本，记录barcode匹配率和未匹配的处理；若数据来自不同批次或不同细胞，说明如何建立共同分析对象。不能以细胞数量相似替代身份对应。样本过滤和细胞群选择会改变调控网络推断，应先固定规则再执行。

检查样本、条件和生物学重复如何进入工作流。一个样本的regulon不能自动代表整个条件；多样本结果需考虑重复和批次。若流程本身不提供所需差异统计，另设合适的统计分析步骤，不将网络边数变化直接当作显著调控差异。

数据库和工作目录的隔离

把SCENIC+代码、数据库和项目输出分成只读/可写目录。每个运行生成单独workdir，避免Snakemake把另一个版本的rule output当成已完成。数据库更新采用新版本目录，先核验散列和读取权限，再切换项目配置；不要覆盖正在运行的共享资源。

多个分析者共用数据库时记录release、物种、基因组版本和维护责任人。对于大型数据库，在工作流启动前检查空间配额和缓存策略；文件不存在或网络下载失败时，错误应在预检查阶段暴露。把临时下载放入明确目录，避免散落在用户家目录并难以清理。

报告模型输出的证据等级

列出从表达和可及性到candidate enhancer、motif及regulon的证据链，并说明哪些环节是预测。候选增强子与基因关联不等于已建立的物理互作；结合预测不等于直接TF实验。将网络输出与差异表达、细胞类型背景和外部验证分别呈现。

如果网络只在某一种数据库或阈值下出现，应报告这种依赖性。对重要候选调控关系保留其峰位置、motif ID、目标基因和运行版本，方便其他人做基因组浏览器复核和实验设计。

运行前团队对照会

正式大样本运行前由分析者和实验者共同确认RNA/ATAC配对、样本ID、细胞过滤、峰集、数据库和目标问题。把决定写进配置和manifest，避免软件可运行就默认生物学输入正确。记录环境、参考、参数与资源边界后再提交作业。

若环境安装复杂，优先使用项目锁定的conda环境或官方建议容器，并保存镜像/环境摘要。容器也不能固定宿主机内核、存储、调度器、GPU驱动或外部数据库内容；这些系统层条件仍需记录。

配套练习包

本PDF对应的ZIP包含本文代码块、主题练习模板或合成数据、README和SHA256校验文件。请从同一教程页面下载练习包；先核对校验值，再在独立目录复现。

参考资料

SCENIC+官方安装入口 · 来源资料

当前固定修订Python与依赖元数据 · 来源资料

SCENIC+ Snakemake官方教程 · 来源资料

固定修订CLI实现 · 来源资料

操作建议：先在合成小样本上验证流程，再依据项目参考、标签合同、存储策略与获配资源验收真实数据。