

浙江大学实验报告

课程名称: 生态学基础及实验 指导老师: 蒋明凯 实验类型: 干实验
实验名称: 生态系统过程模型原理及操作 成绩: _____ 组员: _____

专业: 生物科学
姓名: 蒋贤迪
学号: 3240105782
日期: 2025.12.17
地点: 生物实验中心-312

一、实验目的和要求

1. 实验目的

- 学习生态系统过程模型的基本原理
- 掌握生态系统过程模型 GDAY 的基本操作方法

2. 实验要求

字数不少于 1200 字, 实验报告包括:

- (1) 模型介绍
- (2) 实验方法
- (3) 结果描述
- (4) 讨论

其中, 结果部分不少于 5 副图/表, 包括简要图例。

二、实验内容和原理

1. 生态系统过程模型简介

(1) 过程代表性

生态系统模型根据其侧重的核心生态学过程不同，可分为多种类型。从关注元素循环的生物地球化学模型，到关注植被动态的林隙模型，不同种类的模型在核心要素和应用场景上存在显著差异。各类主要模型的定义、核心要素及典型实例如表 1 所示。

Table 1: 生态系统过程模型的分类及其核心要素

模型种类	核心要素	模型例子
生物地球化学循环模型	模拟生物地球化学循环 (C、N、P)	TEM, CASA, BIOME-BGC, CENTURY, GDAY
林隙模型 (Gap model)	单株树、种群动态、群落组成	JABOWA, FORET, GDAY
生态系统种群结构模型	与林隙模型类似，但基于网格或地块	ED
全球动态植被模型	生物地球化学循环、群落组成、全球生物地理	IBIS, LPJ, SDGVM, SEIB-DGVM
陆面模式	用于天气和气候模拟的全球地表模型，关注水文气象过程与大气的生物地球物理耦合	CLM, JULES, CABLE, JS-BACH
植物冠层模型	多层冠层尺度模型，注重耦合叶片生理过程和冠层物理	MAESPA, CUPID, CANOAK
生态水文模型	与陆面模式类似，拥有水文的横向流动连通性	RHESSys

(2) 空间代表性

除过程机制外，模型所代表的空间尺度也是其重要的分类依据。生态学过程跨越了从微观到宏观的多个层级，模型也相应地被设计用于模拟从单一器官（如叶片）到整个生物圈的不同空间尺度。根据空间代表性的不同，常见模型可分为器官模型、单株模型、群落模型直至全球模型，具体分类及实例如表 2 所示。

Table 2: 不同空间尺度下的生态模型分类及实例

空间尺度/模型类型	典型模型实例
器官模型（叶片或冠层）	Farquhar's model, FOREST-BGC, MAESTRO, BIOMASS
单株生理生态模型	ECOPHYS, TREGRO, TREE-BGC
群落模型	JABOWA, FORET, ZELIG, LINKAGE
林分或生态系统模型	GDAY, PnET, CENTURY, TRIPLEX
景观模型	FIRE-BGC, LANDIS
全球模型	BIOME3, MAPSS, IBIS

(3) 经验模型

经验模型的典型代表是第一个全球净生产力 (NPP) 模型——MIAMI 模型 (Lieth, 1975)。该模型具有以下特征：

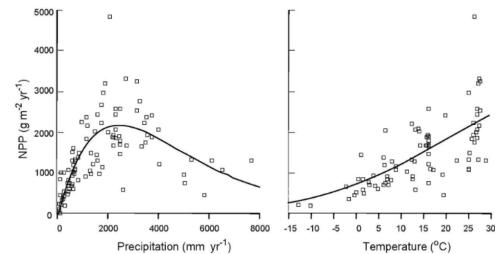
- 数据基础：**量化了 52 个站点的 NPP 与站点年均温和年均降水的关系。

- 计算公式：**

$$NPP_T = \frac{3000}{1 + e^{1.315 - 0.119T}}$$

$$NPP_P = 3000 * (1 - e^{-0.000664P})$$

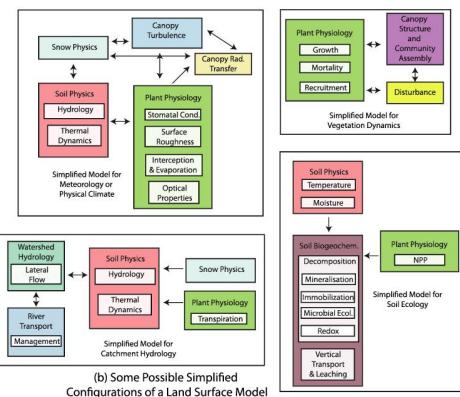
- 预测能力：**预测了全球陆地净生产力 ($63 \times 10^{15} g C yr^{-1}$)。



(4) 过程模型

过程模型描述了生态系统中的主要库 (pool) 和通量 (flux) 以及调节这些变量的因素。它提供了基于基本自然规律和经验关系的预测能力，其核心在于基于过程对生态系统动态及其与环境关系进行理解，主要回答以下问题：

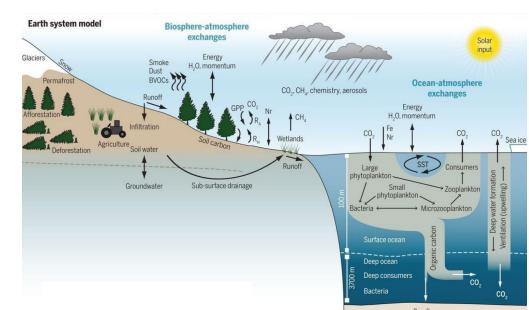
- 光合作用产生的能量？
- 能量或物质如何流动？
- 营养元素在系统中的回收速度有多快？
- 生态系统作为一个整体是如何运作的？
- 系统是如何构造的，它是如何运作的？



(5) 地球系统模型

地球系统模型使用数学公式来模拟驱动地球大气层、水圈、生物圈和地圈的物理、化学和生物过程。其主要特征如下：

- 组成结构：**典型的地球系统模型由大气、海洋、海冰、陆地和冰川的耦合模型组成。
- 土地模块：**以其生态系统、流域、人和环境变化的社会经济驱动因素为代表。
- 功能目标：**全面量化人类和生态系统影响、适应和缓解全球环境变化的过程。



2. 生态系统的定量描述

(1) 模型的核心要素

任何生态系统过程模型，无论其复杂程度如何，都由一组核心要素构成。这些要素定义了模型的边界、内部状态以及与外界环境的交互方式。根据 Bonan (2019) 的总结，生态系统模型的主要组成要素及其定义如表 3 所示。

Table 3: 生态系统过程模型的核心要素定义

要素	定义
系统边界	界定系统内容；将系统内部的过程与系统外部的过程分开
状态变量	作为时间导数出现在预测方程左侧的时变量（如生物量库大小）
模型结构	用于制定模型变量和过程的方程；变量和过程之间的相互作用和联系
参数	过程参数化中使用的常数（如最大光合速率）
输入变量	演化模型状态所需的数量（如气象驱动数据）
初始条件	模型启动需要初始化的状态变量
模型输出	模型模拟的数量；状态变量以及通量和其他诊断变量

(2) 模型发展的重要挑战

尽管生态模型已取得了长足发展，但在准确预测未来全球变化响应时仍面临诸多挑战：

- (1) **过程的代表性与普适性：**如何在模型中不断添加新的生态过程（如磷循环、微生物机制），同时保持模型的普适性。
- (2) **时空尺度问题：**如何将叶片尺度（秒/厘米）的观测数据推演到全球尺度（年/公里）的预测。
- (3) **复杂反馈关系：**例如在 CO_2 升高背景下，植物的气孔响应、养分限制与土壤反馈之间的复杂互作。
- (4) **不确定性：**包括模型内部结构的不确定性（如参数敏感性）和外部环境（如气候情景）的不确定性，以及“异参同效”（Equifinality）带来的验证难题。

3. 教学模型——GDAY

(1) 模型介绍

GDAY (Generic Decomposition And Yield) 模型是一个基于过程的林分尺度生态系统模型，由 Comins 和 McMurtrie (1993) 最初开发。该模型通过耦合林分生长模型与 CENTURY 土壤有机质分解模型，能够模拟生态系统中的碳 (C)、氮 (N)、磷 (P) 和水循环动态。

该模型具有以下主要特征：

- **时间步长:** 支持日步长 (Daily) 或亚日步长 (Sub-daily, 30 分钟) 运行。在亚日步长下，模型采用双叶模型 (Two-leaf approximation) 计算光合作用。
- **核心过程:** 集成了 Farquhar 光合作用模型、基于 PIPE 理论的碳分配机制、以及 CENTURY 模型的土壤碳氮磷周转过程。
- **应用场景:** 广泛用于评估环境变化（如大气 CO_2 浓度升高、氮沉降增加）对森林生产力、水利用效率及养分循环的长期影响。

(2) 模型结构

GDAY 模型的内部结构描述了物质在植被与土壤不同库之间的流动与转化，如图 4 所示。

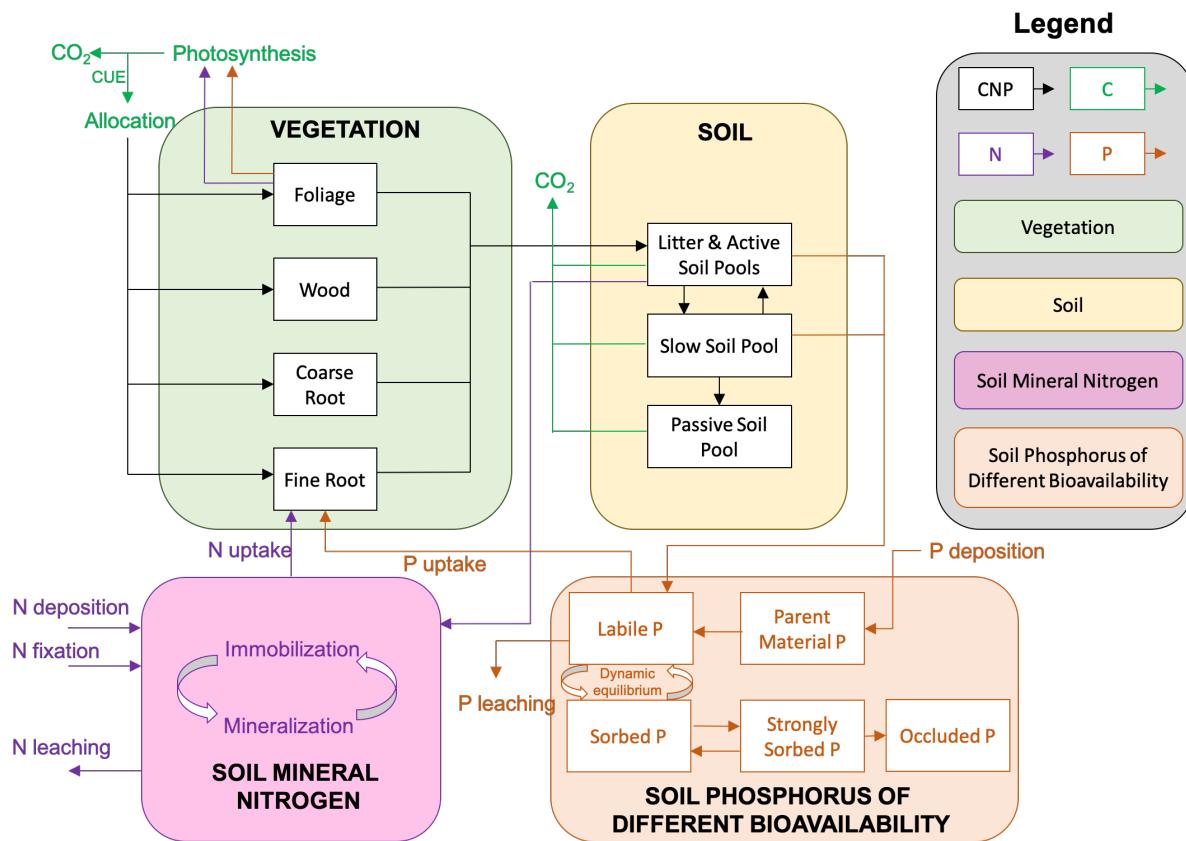


Figure 1: GDAY 模型框架与碳氮磷循环路径示意图（图源 GDAY-CNP 的 README.md）

其核心组成部分及通量路径如下：

- **VEGETATION (植被):** 包含四个主要的生物量库：叶 (Foliage)、木 (Wood)、粗根 (Coarse Root) 和细根 (Fine Root)。模型模拟光合作用产生的碳 (GPP) 在这些器官间的分配、生长以及随后的凋落过程。
- **SOIL (土壤):** 主要指土壤有机质 (Soil Organic Matter) 部分。依据 CENTURY 模型，划分为三个周转速率不同的库：活性库 (Active)、慢速库 (Slow) 和惰性库 (Passive)，模拟凋落物分解及有机质的矿化与固定。

- **SOIL MINERAL NITROGEN (土壤矿质氮)**: 代表土壤中可供植物吸收利用的无机氮库。它接收来自大气沉降、生物固氮及有机质矿化的氮，同时通过植物吸收和淋溶流失。
- **SOIL PHOSPHORUS OF DIFFERENT BIOAVAILABILITY (不同生物有效性的土壤磷)**: 精细刻画了磷的地球化学循环。包含母质磷 (Parent Material P)、易利用磷 (Labile P)、吸附态磷 (Sorbed P)、强吸附态磷 (Strongly Sorbed P) 和闭蓄态磷 (Occluded P)。
- **通量图例说明 (Legend)**: 图中不同颜色的箭头和边框代表不同的物质流向：
 - CNP (黑色): 代表以固定化学计量比流动的有机物质 (包含碳、氮、磷)，如生物量生长和凋落物形成。
 - C (绿色): 代表纯碳通量，主要是光合作用 (CO_2 固定) 和呼吸作用 (CO_2 释放)。
 - N (紫色): 代表氮素通量，包括氮沉降、固氮、矿化、固定、淋溶和植物吸收等过程。
 - P (橙色): 代表磷素通量，包括磷的生物化学矿化、吸附/解吸以及风化输入等过程。

(3) 模型文件结构

GDAY 模型的核心代码库采用模块化的文件结构，将源代码、头文件、参数文件和辅助脚本严格分离。根据课件描述，其主要文件结构如表 4 所示。

Table 4: GDAY 模型核心文件结构及说明

目录类别	文件夹/路径	包含文件及功能说明
源代码	code/src	包含 GDAY 模型源代码，如 <code>gday.c</code> , <code>canopy.c</code> , <code>soils.c</code> 等
头文件	code/src/include	包含 <code>gday.h</code> , <code>canopy.h</code> 等头文件，定义数据结构与函数声明
参数文件	code/params	存放启动模型所需的基本参数文件，如 <code>best_start.cfg</code>
运行代码	code/scripts	包含启动模型的运行代码，如 <code>adjust_gday_param_file.py</code>
描述性文件	code/doc	存放 <code>outline.png</code> ，如图所示

(4) par.cfg 文件

`par.cfg` 是控制模型行为的核心配置文件，主要包含以下部分：

- **[files]**: 设置文件路径。
- **[params]**: 模型模拟所需的相关参数，即调控植物生理、生长、生态系统生物地球化学循环过程的相关参数。
- **[state]**: 模型的状态变量，主要是碳氮磷库信息，用以初始化模型
- **[control]**: 控制模型模块的参数，比如是否输出数据，是否打开 spin-up，应用哪个具体功能模块等
 - `spin_up`: 是否开启预热平衡模拟。
 - `alloc_model`: 碳分配模式 (固定比例 Fixed 或异速生长 Allometric)。

- `textttfixleafnc`: 是否固定叶片碳氮比。
- `[print]`: 设置需要输出的变量列表。

(5) R 语言文件包

为了简化操作流程，本实验通过一套 R 语言文件包来管理 GDAY 模型的运行。各文件及其功能说明如表 5 所示。

Table 5: GDAY 模型 R 语言文件包结构说明

文件/文件夹名	主要功能	说明
Run_programs.R	主控制脚本	包含 Windows 版和 Mac 版
Functions	功能函数库	包含 R 语言环境下对模型进行预处理 (pre-processing)、后处理 (post-processing) 的脚本
data	数据文件夹	存放模型所需的原始气象驱动数据
GDAY	模型仓库	包含 GDAY 的源代码和预先编译好的可执行程序
output	输出文件夹	用于存放后处理脚本生成的最终分析结果和图表
R	环境配置	包含 R 语言环境的相关设置文件
simulations	模拟文件夹	<code>PTP.exe</code> : GDAY 在 Windows 环境下的可执行文件 <code>par.cfg</code> : 控制与模型参数文件 <code>EUC_met_*.csv</code> : 处理好的气象驱动数据文件

(6) 功能模块

GDAY 的核心源代码分布在多个 C 语言文件中。根据代码功能逻辑，可将其划分为核心生态生理过程与辅助数据支持两大类，具体文件功能说明如表 ?? 所示。

文件名	功能说明	文件名	功能说明
<code>gday.c</code>	核心主程序	<code>write_output_file.c</code>	生成输出文件
<code>Makefile</code>	生成可执行文件	<code>photosynthesis.c</code>	光合作用计算
<code>initialise_model.c</code>	模型初始化	<code>plant_growth.c</code>	植物生长与碳分配
<code>read_param_file.c</code>	读取模型所需参数	<code>litter_production.c</code>	凋落物产生计算
<code>read_met_file.c</code>	读取气象驱动数据	<code>soils.c</code>	土壤相关代码
<code>utilities.c</code>	模型辅助代码	<code>constants.h</code>	模型常数
<code>nrutil.c</code>	数值计算与稳定性	<code>structures.h</code>	模型参数的定义
<code>write_output_file.c</code>	生成输出文件		

4. 实验参数讲解

(1) data_model_intercomparison

本部分主要对模型输出结果与观测数据对比分析中涉及的关键生态系统参数进行定义与生态学意义解析。这些参数反映了生态系统的碳库大小、碳流动速率以及植物的生长策略。

• 主要生态系统碳库 (Major Ecosystem Carbon Pools)

该组参数反映碳在生态系统不同组分中的存量（单位为 $g C m^{-2}$ ），是衡量生态系统长期固碳能力的基础。

- **CCR (Carbon Coarse Root):** 粗根生物量碳库，指植物粗根（结构根）部分的碳储量。
- **CFR (Carbon Fine Root):** 细根生物量碳库，指植物细根（吸收根）部分的碳储量。
- **CL (Carbon Leaf):** 叶片生物量碳库，指植物冠层叶片的碳储量。
- **CSOIL (Carbon Soil):** 土壤碳库，指土壤中存在的有机碳总量。
- **CW (Carbon Wood):** 木质部碳库，指树干、树枝等木质组织的碳储量。

• 碳分配系数 (Allocation Coefficients)

该组参数（无量纲，范围 0-1）反映了植物将光合产物 (NPP) 分配给不同器官的策略，体现了植物对环境限制因子的响应。

- **leaf:** 光合产物分配给叶片的比例。
- **root:** 光合产物分配给细根的比例。
- **wood:** 光合产物分配给茎干（木质部）的比例。
- **exudation:** 光合产物分配给根系分泌物的比例。

• 主要碳通量 (Major Carbon Fluxes)

该组参数描述了碳在生态系统与大气之间、以及系统内部流动的速率（单位通常为 $g C m^{-2} yr^{-1}$ ），反映了生态系统的代谢活力。

- **GPP (Gross Primary Productivity):** 总初级生产力。指单位时间内生物（主要是绿色植物）通过光合作用所固定的有机碳量，又称总第一性生产。
- **NPP (Net Primary Productivity):** 净初级生产力。植被所固定的有机碳中扣除本身呼吸消耗的部分，也称净第一性生产力，用公式表示为： $NPP = GPP - R_a$ ，式中， R_a 为自养生物本身呼吸所消耗的同化产物。
- **RAU (Autotrophic Respiration):** 自养呼吸。指植物根、茎、叶等器官为维持生命活动（如维持呼吸、生长呼吸）所消耗的碳。
- **NEP (Net Ecosystem Productivity):** 净生态系统生产力。计算公式为 $NEP = NPP - R_h$ ，其中 R_h 为异养生物呼吸消耗量。它衡量了整个生态系统是作为碳汇（正值，吸收碳）还是碳源（负值，排放碳）。

(2) CO_2 _responses

本部分针对 CO_2 响应分析图表 (`CO2_responses`) 中涉及的关键变量及统计指标进行定义与解释。

- **LAI (Leaf Area Index)**

中文名称：叶面积指数

定义：单位土地面积上植物叶片总面积（单面）的比值（单位： $m^2 \text{ leaf } m^{-2} \text{ ground}$ ）。

模型含义：由叶片碳库 (`shoot`) 通过比叶面积 (SLA) 转换而来。

生态意义：衡量植被茂密程度的核心指标，直接决定光截获能力。

代码源码： `plant_growth.c` 中的 `carbon_allocation` 函数。具体代码见附录。

- **Shoot**

中文名称：叶片生物量碳库 (Foliage Biomass Carbon)

特别说明：特指叶片 (Leaf) 的碳储量，不包含树枝和茎干。

单位： $g C m^{-2}$ 。

生态意义：光合作用的“工厂”，属于周转较快的活性碳库。

代码源码： `plant_growth.c` 中的 `update_plant_state` 函数。

```
1 s->shoot += f->cpleaf - f->deadleaves - f->ceaten
```

- **Stem**

中文名称：茎干生物量碳库 (Wood/Stem Biomass Carbon)

定义：指树干及木质部的碳储量。

模型含义：长周期碳库，用于构建支撑结构。

生态意义：代表森林生态系统的长期固碳能力。

代码源码： `plant_growth.c` 中的 `update_plant_state` 函数。

```
1 s->stem += f->cpstem - f->deadstems
```

- **ET (Evapotranspiration)**

中文名称：生态系统蒸散发

定义：生态系统水分输出的总和，由植物蒸腾 (T)、土壤蒸发 (E_{soil}) 和冠层截留蒸发 (E_{canopy}) 三部分组成，计算公式为 $ET = T + E_{soil} + E_{canopy}$

生态意义：反映生态系统的水分消耗状况。在 eCO_2 下，植物往往通过降低气孔导度减少蒸腾，从而提高水分利用效率 (WUE)。

代码源码： `water_balance.c` 中的 `update_water_storage_recalwb` 函数。

```
1 f->et = f->transpiration + f->soil_evap + f->canopy_evap;
```

三、 实验材料与设备

本实验在安装有 R 语言环境的个人计算机上进行，使用 GDAY 生态系统过程模型（包含 C 语言源代码、编译后的可执行文件 PTP.exe、参数配置文件 par.cfg 及气象驱动数据）完成模拟与分析。

四、 操作方法和实验步骤

本实验在 Windows 操作系统下进行，通过 R 语言脚本 (Run_programs_Windows.R) 调用并控制 GDAY 模型完成全流程模拟。实验步骤严格遵循脚本逻辑，分为前期准备、正式模拟和结果分析三个阶段。

1. 前期准备与数据预处理

- (1) **环境配置：**清理 R 工作环境，加载自定义函数库脚本 R/prepare_R.R，并指定文件输出路径为 simulations 文件夹。
- (2) **程序部署：**调用 copy_gday_executable_to_simulation_folder 函数，将 Windows 版本的 GDAY 可执行文件复制到模拟工作台。
- (3) **气象数据生成：**
 - 调用 create_daily_met_for_GDAY 函数，读取 data/forcing 下的原始数据，准备气象数据。
 - 执行 check_GDAY_met_data_consistency 检查数据一致性。
 - 降水处理：直接调用 change_rainfall 函数，基于原始数据生成降水减半后的气象文件，并存储于目标文件夹。
- (4) **参数初始化：**调用 update_par_cfg_for_spinup 函数修改 par.cfg 配置文件，设置输入气象文件为 EUC_met_spinup.csv，开启 spin_up = TRUE，并指定自旋方法为 BRUTE，为预热模拟做准备。

2. 正式模拟：Spin-up 与实验一（标准情景）

- (1) **模型预热 (Spin-up)：**设置工作路径为 simulations，通过系统命令调用 PTP.exe -s -p par.cfg 运行模型。此步骤模拟工业革命前的平衡状态，并输出 model_spinup.cfg 作为后续模拟的初始状态。
- (2) **环境 CO_2 (aCO_2) 模拟：**调用 update_par_cfg_for_aCO2 函数更新配置，关闭 Spin-up 模式，指定气象文件为 EUC_met_aCO2.csv。运行 PTP.exe -p par.cfg，生成模拟结果 simulation_aCO2.csv。
- (3) **升高 CO_2 (eCO_2) 模拟：**调用 update_par_cfg_for_eCO2 函数，指定气象文件为 EUC_met_eCO2.csv。再次运行模型，生成模拟结果 simulation_eCO2.csv。
- (4) **标准情景结果分析：**模拟完成后，恢复工作环境路径 (setwd(base_dir))，并调用后处理函数对标准情景进行详细评估：

- **质量平衡检查:** 调用 `mass_balance_check`, 验证生态系统内的碳、氮、水闭合情况。
- **CO_2 响应幅度:** 调用 `co2_response_check`, 计算标准情景下的 CO_2 施肥效应。
- **数据验证:** 调用 `data_model_intercomparison`, 将模拟结果与实测数据进行比对。

3. 实验二：降水减半情景模拟

利用预处理阶段生成的修正降水气象数据，重新评估生态系统响应：

- (1) **设置工作路径:** 定义基础路径与模拟文件夹路径，并将 R 工作目录切换至 `simulations` 文件夹以准备运行。
- (2) **aCO_2 模拟:** 调用 `update_par_cfg_for_aCO2`, 指定气象文件为 `EUC_met_aCO2_rev_rain.csv`, 输出文件命名为 `simulation_aCO2_rev_rain.csv`。运行模型。
- (3) **eCO_2 模拟:** 调用 `update_par_cfg_for_eCO2`, 指定气象文件为 `EUC_met_eCO2_rev_rain.csv`, 输出文件命名为 `simulation_eCO2_rev_rain.csv`。运行模型。
- (4) **结果分析:** 恢复工作环境路径至主目录，并调用 `co2_response_check` 函数。特别指定参数 `scenario = "_rev_rain"`, 专门分析降水改变情景下的 CO_2 响应幅度。

4. 实验三：固定叶片 C:N 比情景模拟

- (1) **设置工作路径:** 同样将 R 工作目录切换至 `simulations` 文件夹。
- (2) **aCO_2 模拟:** 调用 `update_par_cfg_for_aCO2`, 恢复使用标准气象文件 `EUC_met_aCO2.csv`, 但新增参数 `fixleafnc = TRUE` 以强制固定叶片碳氮比。输出文件为 `simulation_aCO2_fixleafnc.csv`。运行模型。
- (3) **eCO_2 模拟:** 调用 `update_par_cfg_for_eCO2`, 同样设置 `fixleafnc = TRUE`, 输出文件为 `simulation_eCO2_fixleafnc.csv`。运行模型。
- (4) **结果分析:** 恢复工作环境路径，并调用 `co2_response_check` 函数。特别指定参数 `scenario = "_fixleafnc"`, 分析固定叶片碳氮比情景下的 CO_2 响应幅度。

5. 课后作业

- (1) 在默认参数情境下，结合模型输出结果定量描述大气 CO_2 浓度升高对生态系统植被碳库的影响，并计算 2012-2016 年叶片碳库的 CO_2 施肥效应
- (2) 探究在降水量下降一倍情境下，生态系统植被碳库（分为环境 CO_2 浓度和 CO_2 升高处理）是如何改变的
- (3) 探究在叶片碳氮比固定情境下，生态系统植被碳库如何响应大气 CO_2 升高处理

五、实验结果与分析

1. 数据提取与量化方法

为了从 GDAY 模型输出的逐日时间序列数据中提取关键指标，本实验编写了 R 语言自动化脚本进行数据清洗与统计分析，相关代码见附录。处理流程如下：

(1) **时间窗口筛选**: 选取模拟实验的最后 5 年（2012-2016 年）作为分析窗口，以评估 CO_2 浓度升高后的稳定响应。

(2) **统计指标计算**:

- **通量变量 (Flux)**: 对于总初级生产力 (GPP) 和蒸散发 (ET)，先计算每年的累积总量，再求 5 年平均值。
- **状态变量 (Pool)**: 对于叶面积指数 (LAI) 和茎干生物量 (Stem)，直接计算该时间段内的日均值。

(3) **单位换算与效应计算**: 将模型输出的碳量单位由 $t C ha^{-1}$ 转换为 $g C m^{-2}$ (乘以 100)，并计算 CO_2 施肥效应：

$$\text{Fertilization Effect (\%)} = \frac{eCO_2 - aCO_2}{aCO_2} \times 100\%$$

线
订
装

基于上述方法，三个实验情景下的关键生态系统变量统计结果如表 6 所示。该表格为后续定量分析提供了基础数据支撑。

Table 6: 2012-2016 年间不同实验情景下生态系统关键变量的统计均值与 CO_2 施肥效应

实验情景	变量 (单位)	aCO_2 均值	eCO_2 均值	施肥效应 (%)
Exp 1: 标准情景	GPP ($g C m^{-2} yr^{-1}$)	3949	4961	25.63
	LAI ($m^2 m^{-2}$)	12.66	14.81	16.98
	Stem ($g C m^{-2}$)	10402	11041	6.14
	ET ($mm yr^{-1}$)	1005	991	-1.39
Exp 2: 降水减半	GPP ($g C m^{-2} yr^{-1}$)	2485	3320	33.60
	LAI ($m^2 m^{-2}$)	8.19	9.96	21.61
	Stem ($g C m^{-2}$)	6183	6696	8.30
	ET ($mm yr^{-1}$)	532	531	-0.23
Exp 3: 固定 C:N	GPP ($g C m^{-2} yr^{-1}$)	3949	4961	25.63
	LAI ($m^2 m^{-2}$)	12.66	14.81	16.98
	Stem ($g C m^{-2}$)	10402	11041	6.14
	ET ($mm yr^{-1}$)	1005	991	-1.39

在进行具体的实验情景分析前，本研究首先对模型运行结果进行了质量控制 (Quality Control, QC) 检查。通过验证碳循环 (如 GPP、NPP 与呼吸通量的平衡) 与水循环 (降水、蒸散发与土壤储水量的平衡) 的质量守恒情况，确认模型在模拟期间未出现异常的物质流失或增加 (详细检测结果见附录图 13)。结果表明模型运行稳定，数据质量可靠，可用于后续的定量分析。

2. 实验一：标准情景

在默认参数情境下，本部分首先验证模型对生态系统基准状态的模拟能力，随后分析大气 CO_2 浓度升高对生态系统的具体影响。

(1) 模型基准状态与机制验证

为了确保分析的可靠性，我们将模型在环境 CO_2 下的模拟结果与实测数据进行了对比（图 ??），图中 Sim 代表模型模拟值，Obs 代表实测值。。

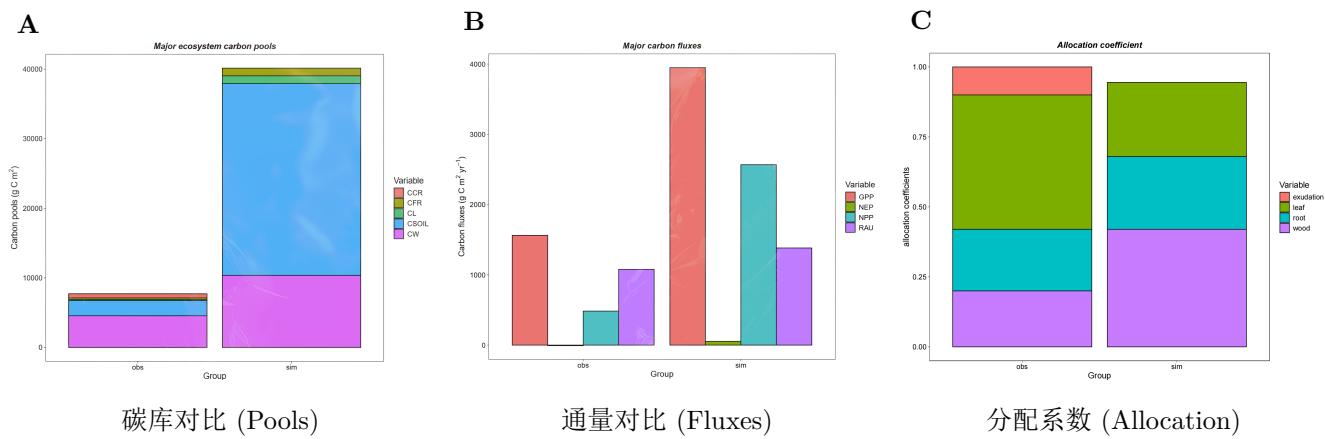


Figure 2: 模型在环境 CO_2 下的基准状态验证: (A) 主要生态系统碳库, (B) 主要碳通量, (C) 碳分配系数。

• 碳库拟合 (图 A):

- 对于植被碳库中最大的组分——茎干 (CW)，实测值 (Obs) 约为 $5000\ g\ C\ m^{-2}$ ，而模型模拟值 (Sim) 约为 $10402\ g\ C\ m^{-2}$ 模型结果显著高于实测值，差异倍数约为 2 倍。同时，土壤碳库 (CSOIL) 的模拟值 (约 $27000\ g\ C\ m^{-2}$) 也远高于实测值 (约 $2000\ g\ C\ m^{-2}$)。
- 分析原因：这种显著的差异表明，默认参数下的 Spin-up 过程使模型达到了一种理论上的高碳平衡态，而现实中的生态系统可能受限于树龄、化学污染等因素，尚未积累如此多的生物量。

• 通量对比 (图 B):

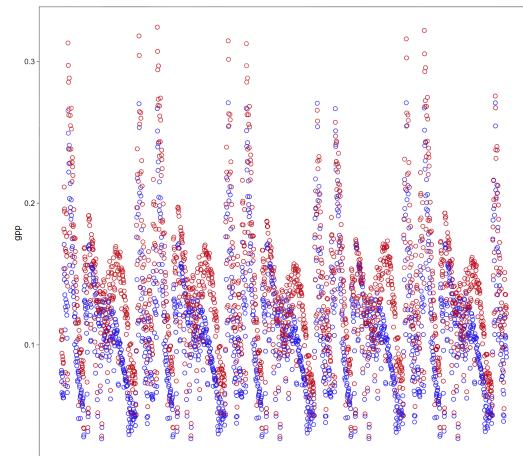
- 模型模拟的总初级生产力 (GPP) 为 $3949\ g\ C\ m^{-2}\ yr^{-1}$ ，而实测值约为 $1500\ g\ C\ m^{-2}\ yr^{-1}$ ，模型结果显著高于实测值。类似的，生态系统呼吸 (RAU) 的模拟值也高于实测值。
- 分析原因：通量的高估直接解释了为何图 A 中的生物量积累会被高估，因为碳输入的源头 (GPP) 设定偏高。

• 分配策略 (图 C):

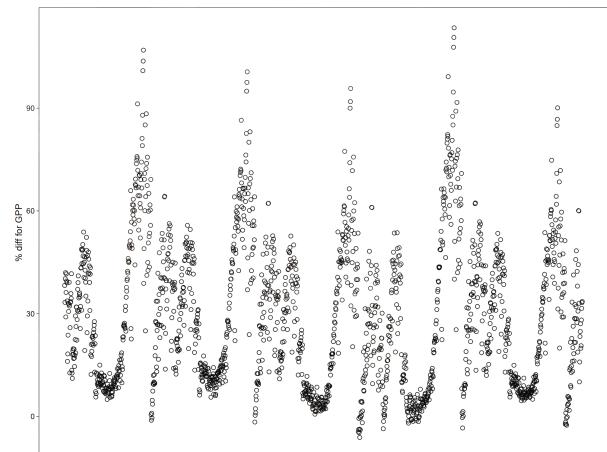
- 在光合产物的分配上，实测数据显示植物将约 50% 的碳分配给叶片 (leaf)，但模型仅分配了约 30%，同时茎秆 (wood) 的分配显著高于实际值，约为实际的 2 倍。
- 分析原因：这说明模型采取了更倾向于茎干的分配策略，而不是叶片。这种分配机制的偏差会进一步影响模型对未来 LAI 和光合作用的预测。

(2) 大气 CO_2 浓度升高对生态系统的影响

(i) 总初级生产力 (GPP)

A

GPP 时间动态

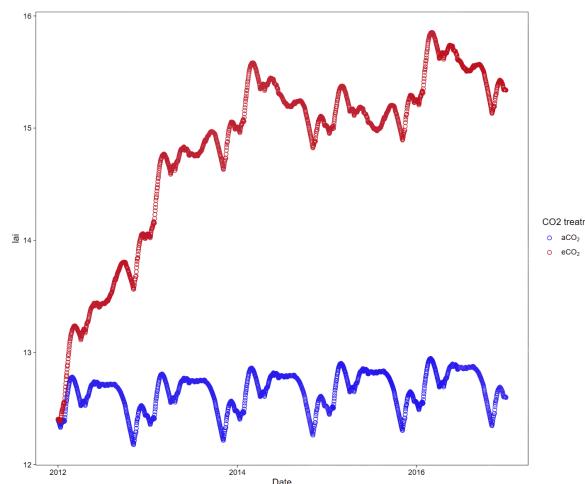
B

GPP 增加幅度 (%)

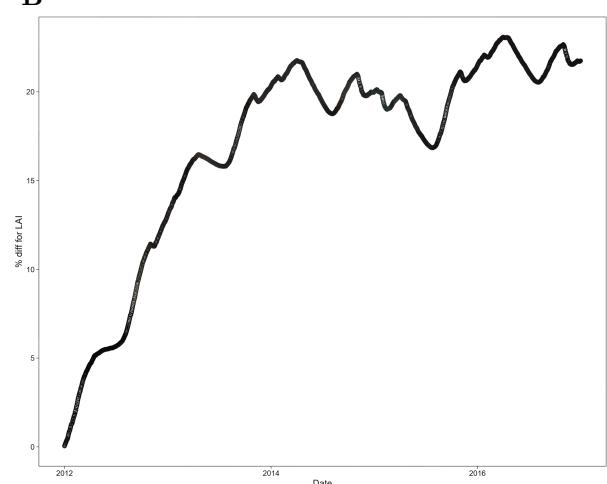
Figure 3: 光合碳获取对 eCO_2 的响应线
订
装
线

如图 3 A 所示， eCO_2 处理（红线）下的 GPP 常常高于环境 CO_2 组。根据图 B 的统计，在 2012-2016 年间，GPP 的增加幅度（施肥效应）主要集中在 10 ~ 50% 的区间内，平均增幅约为 25.6%，且在每一年中间都有一个明显的低谷，大约 10%。这表明高浓度 CO_2 解除了 Rubisco 酶的限制，直接促进了碳同化速率。

(ii) 叶片碳库与叶面积指数 (LAI)

A

LAI 时间动态

B

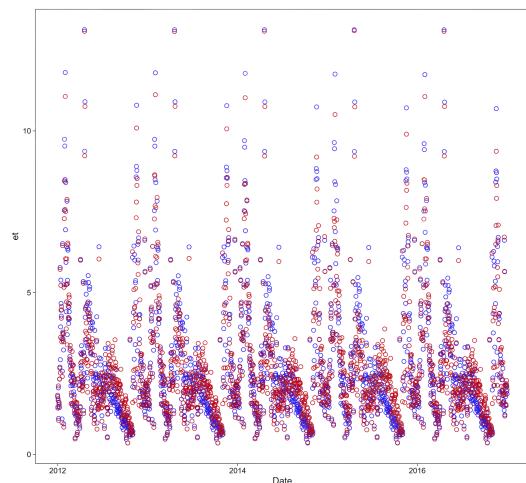
LAI 增加幅度 (%)

Figure 4: 冠层结构（叶碳库）对 eCO_2 的响应

图 4 A 显示， eCO_2 下的 LAI 始终高于 aCO_2 处理，平均增加幅度约为 17.0%。再看图 4 B，该图展示出了 2012-2016 年间叶片碳库的 CO_2 施肥效应。从 2012 年到 2014 年，增肥效应总体上不断上升，但在之后逐渐出现波动甚至减少，但在到达 2016 年后又回到了和 2014 年相近的水平。这意味着光合产物的增加不仅用于呼吸消耗，还成功转化为了冠层结构的扩增，只不过随着时间的增长，该扩增作用并不稳定，可能受到其他限制因素的影响。

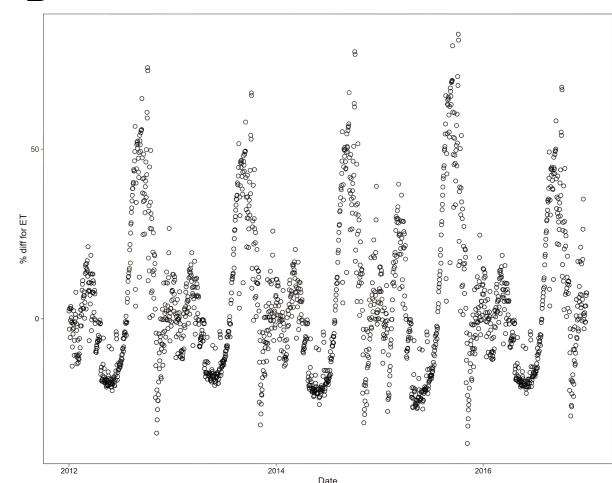
(iii) 蒸散发 (ET)

A



ET 时间动态

B



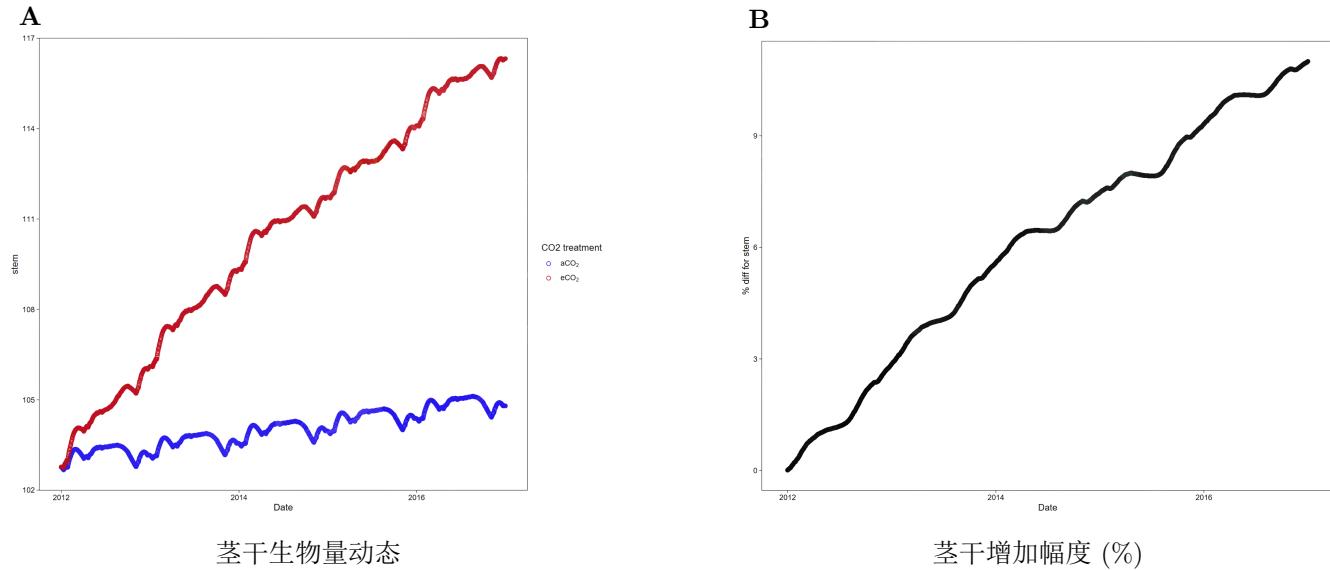
ET 变化幅度 (%)

Figure 5: 水分利用对 eCO_2 的响应线
订
装

与其他变量不同， eCO_2 与 aCO_2 处理下的 ET 无明显差异（见图 5），平均差异仅为 -1.4%。从图 A 中可以看出，红色和蓝色数据点并未出现明显的差异，而图 B 也展示出了 ET 的变化幅度是时正时负的。 CO_2 的增肥效应总体表现为在一年的周期中，先小幅度增加到约 20%，再降低到约-20%，然后又回升到 70%，又下降到约-35%，最后回到约 0%。这种现象表明，尽管叶面积增加了，但由于植物气孔的调节作用，整体蒸散发并未显著增加，且波动较大。

(iv) 茎干 (Stem)

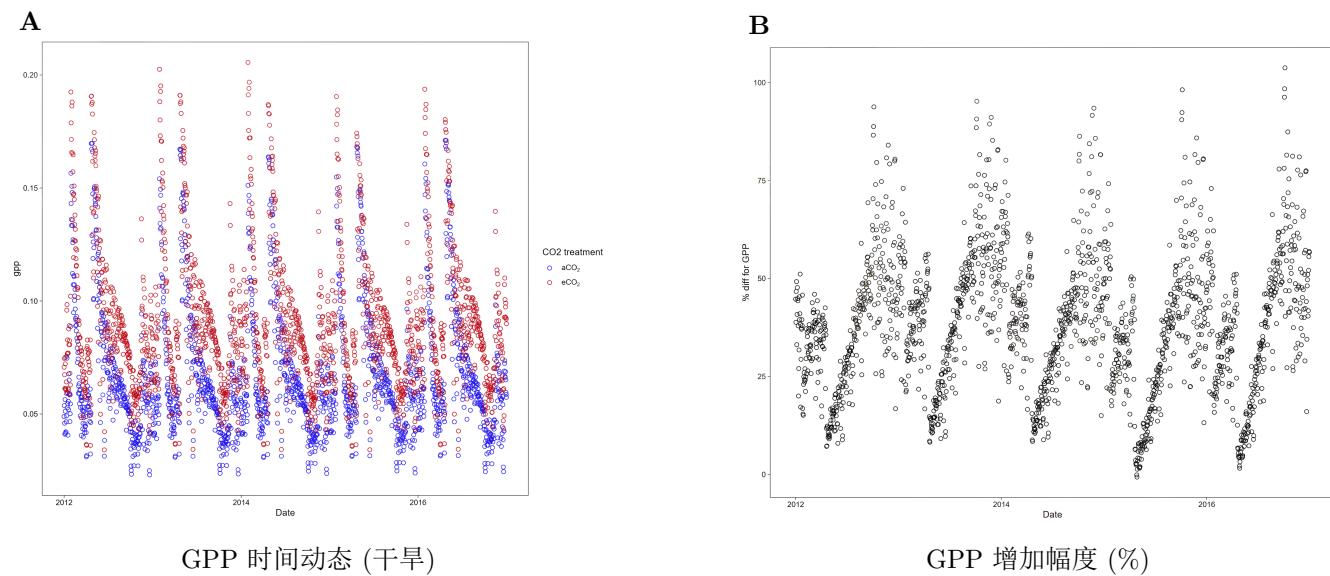
茎干是长期碳汇的主要载体。如图 6A 所示， eCO_2 的茎干生物量始终高于 aCO_2 ，而图 B 显示施肥效应随时间推移大致呈线性上升趋势。至 2016 年末，茎干碳库在 eCO_2 下比对照组增加了约 6.1%。这证明了 CO_2 施肥效应具有累积性，额外的碳被有效固定在了长周转周期的木质部中。

Figure 6: 茎干碳库对 eCO_2 的累积响应

3. 实验二：降水减半情景

本实验旨在探究降水减少 50% 的情况下，生态系统植被碳库如何改变。重点分析在干旱胁迫下，生态系统的绝对生产力是否下降，以及 eCO_2 是否通过提高水分利用效率表现出更强的施肥效应。

(i) 总初级生产力 (GPP)

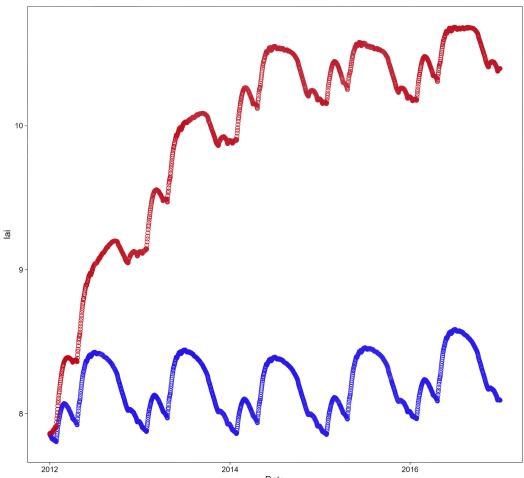
Figure 7: 降水减半情景下，GPP 对 eCO_2 的响应

根据图 7 并结合实验一结果分析：

- 干旱胁迫效应（与实验一对比）：**受水分限制影响，实验二中 aCO_2 处理下的 GPP 年均值仅为 $2485 g C m^{-2} yr^{-1}$ ，相比实验一的标准情景 ($3949 g C m^{-2} yr^{-1}$) 下降了约 37%。表明降水减半严重抑制了生态系统的光合生产力。
- 相对施肥效应增强：**观察图 B，在干旱条件下， eCO_2 带来的 GPP 增加幅度（施肥效应）平均达到了 **33.6%**。
- 结论：**与实验一（平均增幅 25.6%）相比，干旱情景下的 CO_2 施肥效应显著**增强**。这说明 eCO_2 在水分受限时对植物的促进作用更为关键，体现了高浓度 CO_2 对干旱胁迫的缓解作用。

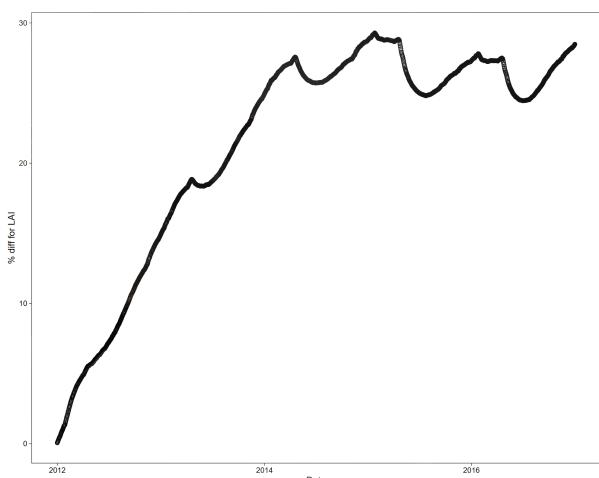
(ii) 叶片碳库与叶面积指数 (LAI)

A



LAI 时间动态 (干旱)

B



LAI 增加幅度 (%)

Figure 8: 降水减半情景下，LAI 对 eCO_2 的响应

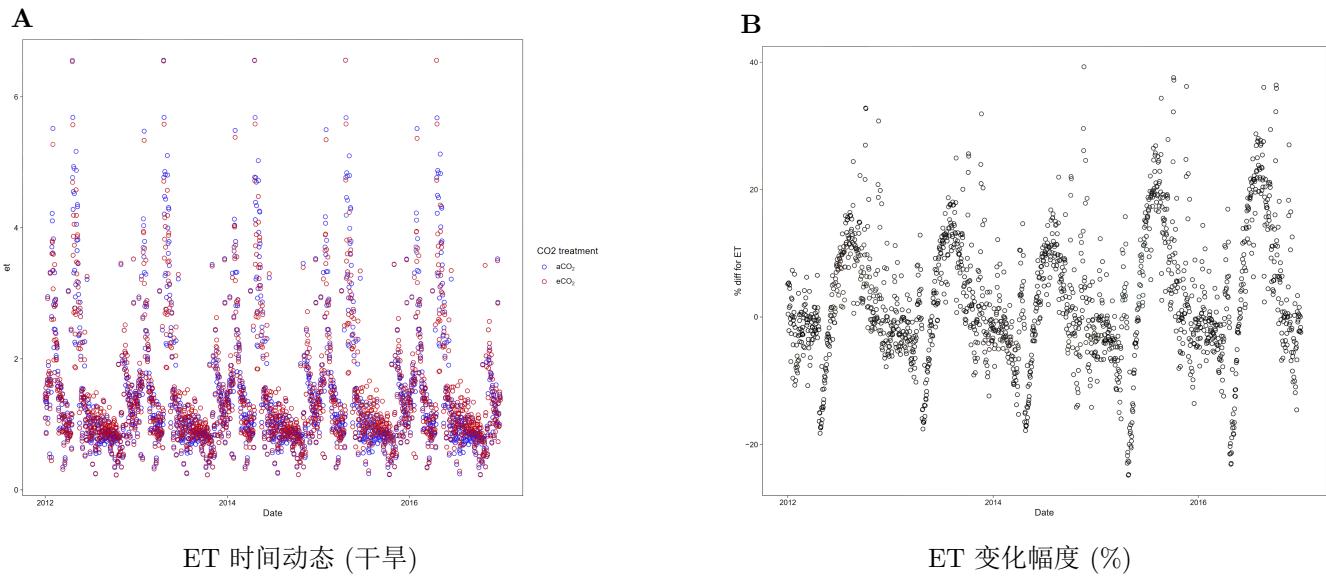
- 结构受损：**对比图 8 A 与实验一，干旱导致 aCO_2 下的 LAI 均值从实验一的 12.66 下降到了 8.19。这符合植物在缺水时通过减少叶面积来降低蒸腾耗水的生存策略。
- 恢复能力：**图 B 显示，在 eCO_2 作用下，LAI 的相对增加幅度约为 **21.6%**，高于实验一的 17.0%。这表明虽然干旱限制了总叶面积，但高浓度 CO_2 依然支持植物维持相对更多的冠层结构。

(iii) 蒸散发 (ET)

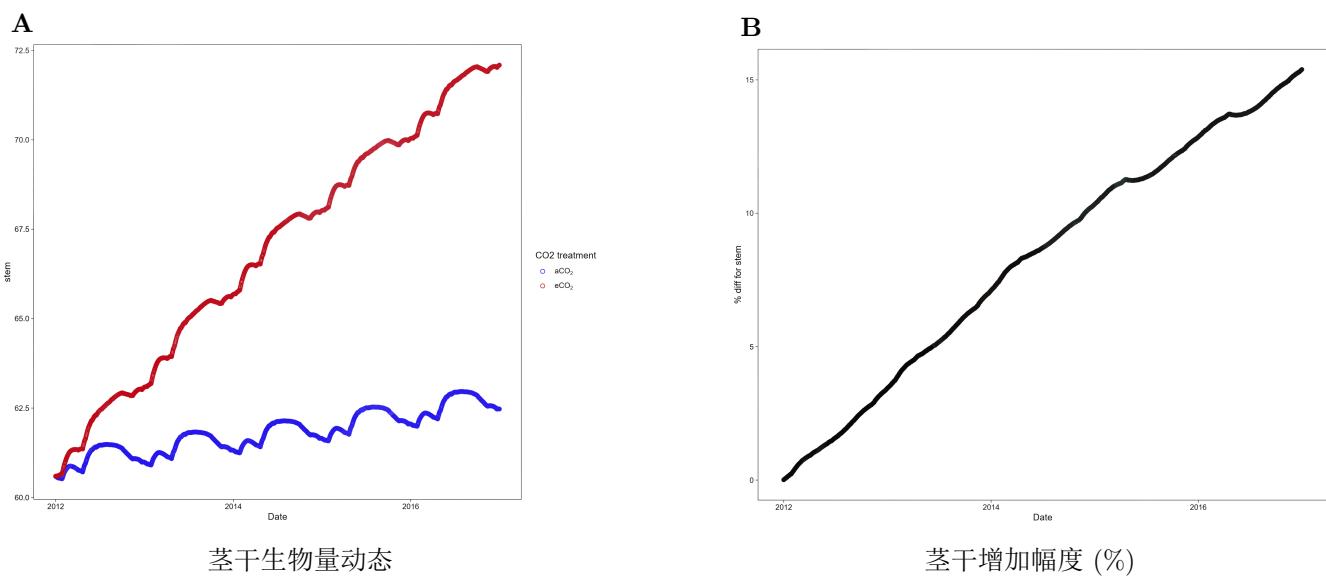
本环节是解释实验二结果的关键机制：

- 如图 9 A 所示，由于水分供应减半，ET 的年均值降至 $532 mm yr^{-1}$ (实验一约为 $1005 mm yr^{-1}$)，绝对值大幅下降。
- 观察图 B (eCO_2 效应)，ET 的变化幅度表现为微弱的负值 (-0.2%)。

- 机制分析：**虽然 GPP 大幅增加了 33.6%，但水分消耗 (ET) 几乎没有增加。这意味着水分利用效率 (WUE = GPP/ET) 得到了极大提升。在干旱情景下， eCO_2 诱导的气孔关闭成为了植物生存的关键机制，使其用同样甚至更少的水换取了更多的碳。

Figure 9: 降水减半情景下，ET 对 eCO_2 的响应

(iv) 茎干 (Stem)

Figure 10: 降水减半情景下，茎干碳库对 eCO_2 的响应

- 积累受阻：**图 A 显示，受干旱限制，茎干生物量的年均值约为 6183 g C m^{-2} ，显著低于实验一 (10402 g C m^{-2})，表明生长速度严重放缓。

- **施肥效应：**图 B 显示，2016 年末茎干碳库的 CO_2 施肥效应约为 8.3%。对比实验一的数值（6.1%），干旱情景下的相对增幅更高。这有力地证明了大气 CO_2 浓度升高能显著缓解干旱对森林生长的负面影响，使植被在水分受限环境中仍能保持一定的碳积累能力。

4. 实验三：固定叶片 C:N 比情景

(1) 实验结果描述

在 R 脚本中设定 `fixleafnc = TRUE` 并运行模型后，我们发现实验三的所有输出结果与实验一完全一致，没有任何数值上的差异。

“固定” C:N 比前后产生相同结果这一反直觉的现象促使我们深入探究模型的底层运行机制。通过代码溯源与数据验证，我们证实了这是由于生态系统处于极端氮限制状态，导致实验一中的“可变 C:N 比”实际上已被锁定在生理下边界上。

(2) 机理探究

为了解释这一现象，我们结合了 C 语言源代码、参数配置文件及模拟输出数据进行了详细排查（图 11）。

(i) 理论边界 (Boundary):

检查模型基础配置文件 `base_start_with_P.cfg`，发现参数 `nf_min`（叶片最小氮浓度）被设定为：`nf_min = 0.005` 这对应着叶片 C:N 比的生理上限为 200 (1/0.005)。

同时，检查另一基础配置文件 `par.cfg`，发现参数 `ncmaxfold` 和 `ncmaxfyouth` 被设定为 0.04，这对应着叶片 C:N 比的生理下限为 25 (1/0.04)

(ii) 代码限制:

查阅源代码 `plant_growth.c` 中的 `calculate_growth_stress_limitation` 函数，模型定义了如下逻辑：当叶片氮浓度 (`shootnc`) 降至 `nf_min` 以下时，氮限制因子 (`nlim`) 将变为 0，从而强制停止生长。具体代码是：

```
1 if (s->shootnc < p->nf_min) {
2     nlim = 0.0;
3 }
```

而在同一代码的 `update_plant_state` 函数中，也以相似的逻辑确定了上边界，不过在本次实验中该函数未发挥作用。

(iii) 数据验证 (Data Verification):

为了确定是上边界还是下边界，我编写了 R 验证脚本，提取实验一输出文件中的叶片氮含量 (`shootn`) 与碳含

量 (shoot)，计算其实际比值。如图所示，在 2012-2016 年间，实验一的叶片 N:C 比并非自由波动，而是始终被“锁定”在 0.005 的直线上。

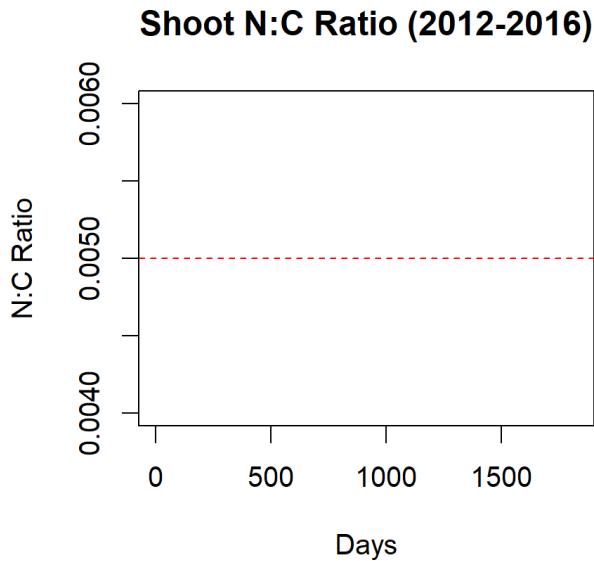


Figure 11: 实验一模拟期间叶片 N:C 比的时间序列验证图。红虚线代表参数文件设定的下边界 (0.005)。

线
订
装

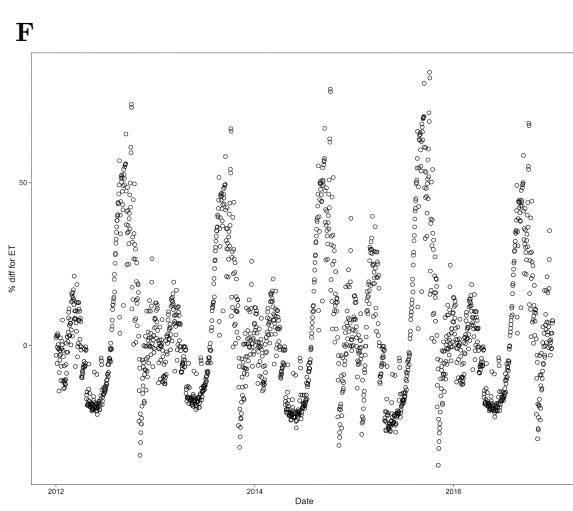
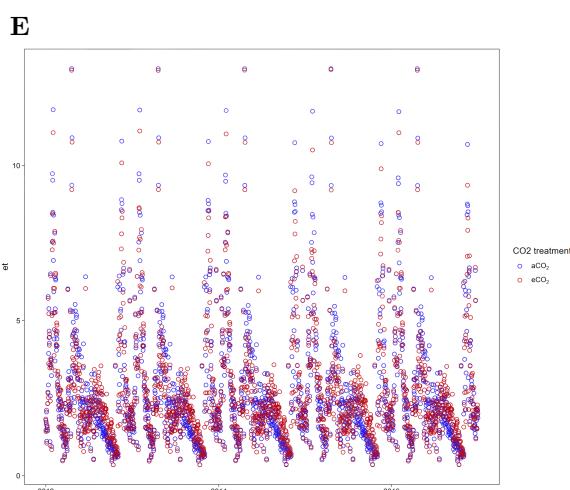
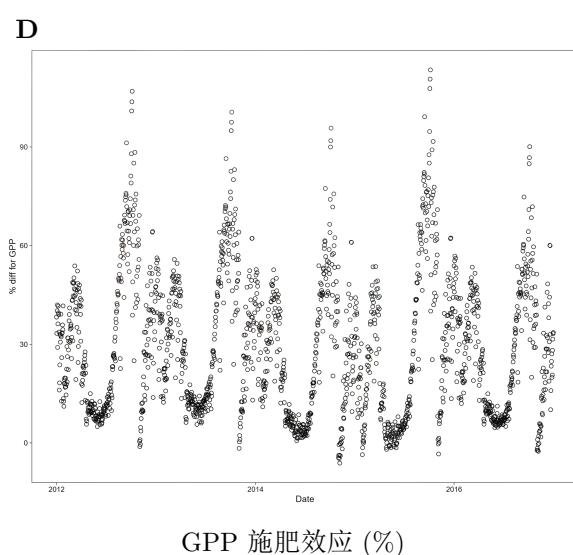
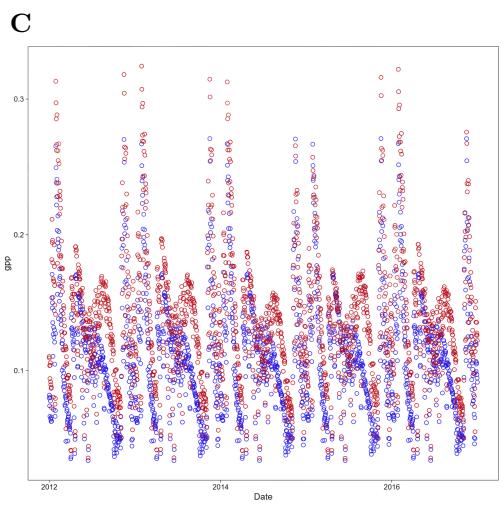
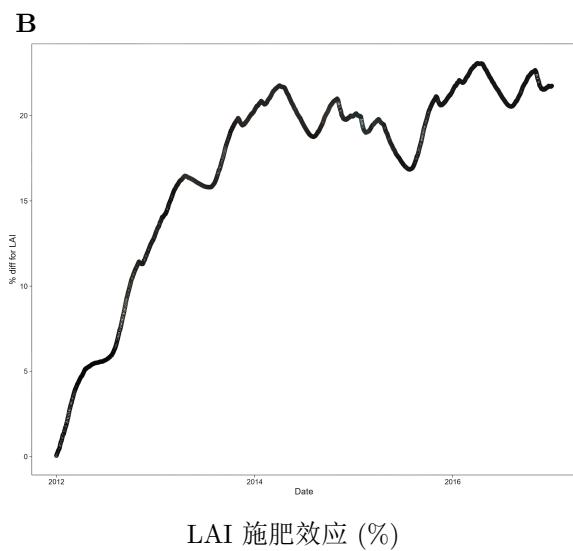
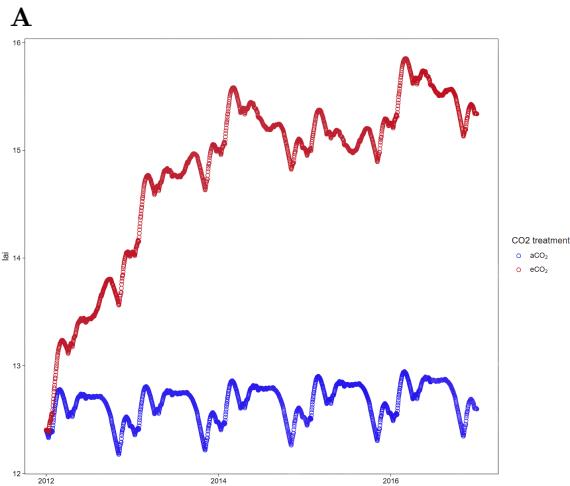
结论：

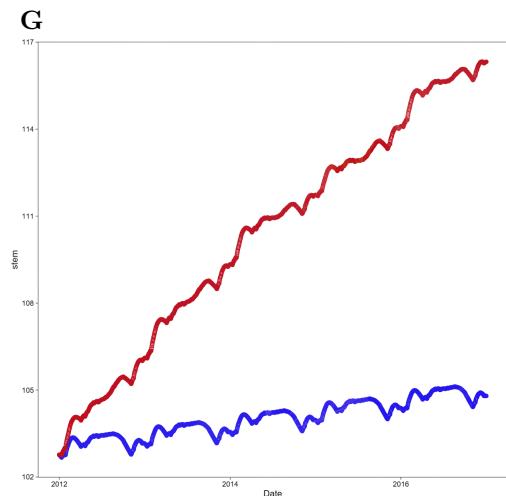
对比实验三与实验一的结果可知，在未设定固定 C:N 比时，模型在模拟时，C:N 比是在一个设定边界里变化的。既然我们在实验三中固定了 C:N 比后结果没有发生变化，验证结果表明，在实验一的模拟中，叶片 N:C 比已持续处于参数设定的下限值。因此，固定 C:N 比并未改变模型的运行状态，导致两实验结果一致。如图 11 的验证所示，这一边界即为 `nf_min = 0.005`。由于氮的限制，实验一中的变量被迫触及并停留在了这一生理下限；因此，当实验三固定 C:N 比时，实际上只是延续了这一已存在的边界状态，从而导致两种情景下的模拟结果完全一致。

注：由于实验三的生态系统响应图表与实验一完全相同，为了实验报告的美观，相关输出图片在附录中展示，仅在正文展示 C:N 比的验证结果。验证用 R 代码详见附录。

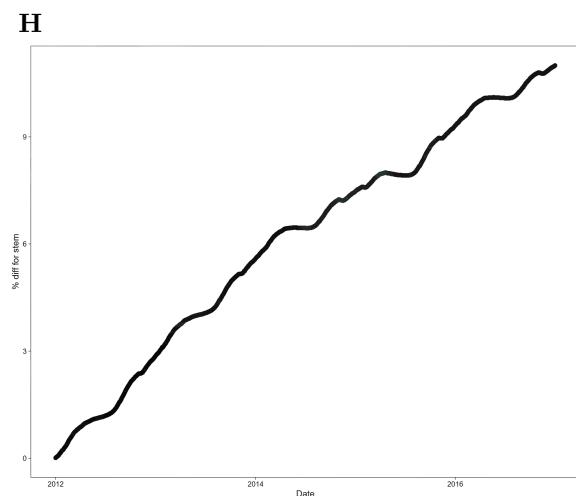
六、附录

1. 实验三相关图片

线
订
装
线



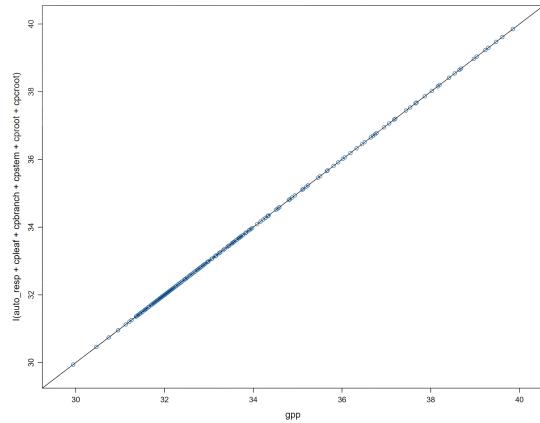
Stem 生物量动态



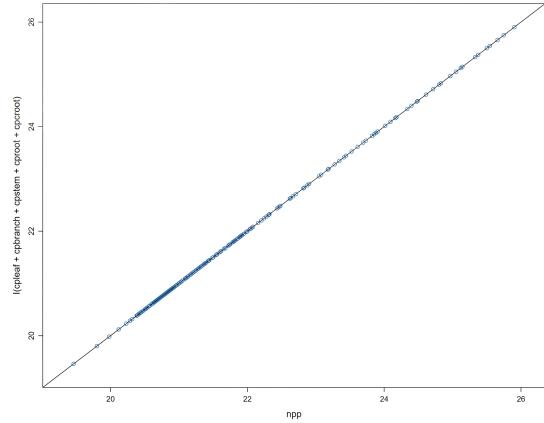
Stem 施肥效应 (%)

Figure 12: 实验三（固定叶片 C:N 比情景）关键生态系统变量对 eCO_2 的响应。

2. 模型运行质量控制检查

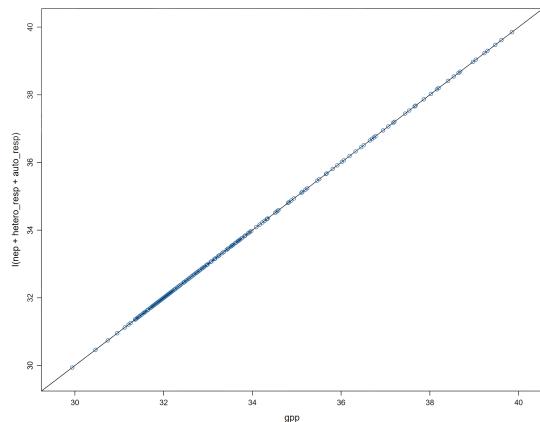


碳通量平衡检查 A

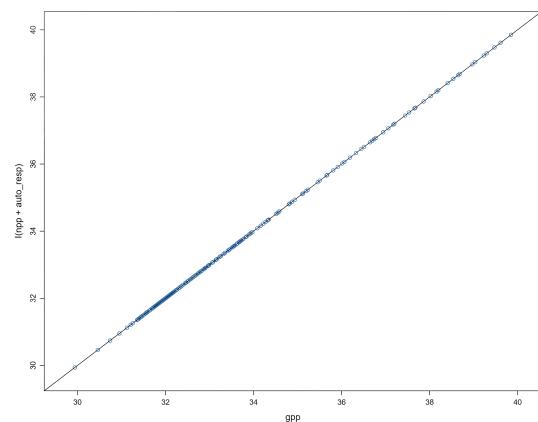


碳通量平衡检查 B

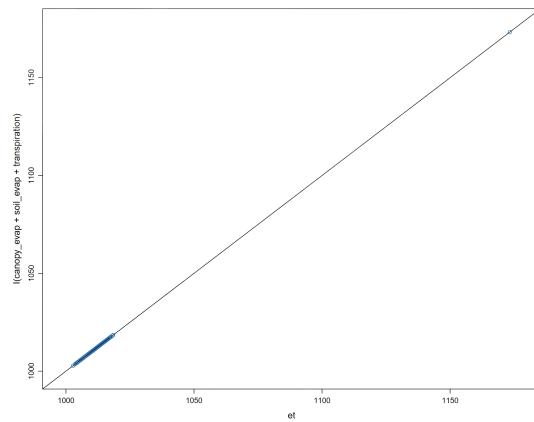
线
订
装



NEP 与呼吸平衡检查



NPP 与呼吸平衡检查



水循环质量守恒检查

Figure 13: 模型碳、水循环质量守恒验证结果。图中残差（Residual）接近于 0 表明模型满足物质守恒定律。

3. 相关代码

Listing 1: 更新叶面积指数 (LAI) 的 C 代码片段

```

1  if (c->deciduous_model) {
2      if (float_eq(s->shoot, 0.0)) {
3          s->lai = 0.0;
4      } else if (s->leaf_out_days[doy] > 0.0) {
5          s->lai += (f->cpleaf *
6              (p->sla * M2_AS_HA / (KG_AS_TONNES * p->cfracts)) -
7              (f->deadleaves + f->ceaten) * s->lai / s->shoot);
8      } else {
9          s->lai = 0.0;
10     }
11 } else {
12     /* update leaf area [m2 m-2] */
13     if (float_eq(s->shoot, 0.0)) {
14         s->lai = 0.0;
15     } else {
16         s->lai += (f->cpleaf *
17             (p->sla * M2_AS_HA / (KG_AS_TONNES * p->cfracts)) -
18             (f->deadleaves + f->ceaten) * s->lai / s->shoot);
19     }
20 }
21
22 if (c->fixed_lai) {
23     s->lai = p->fix_lai;
24 }
```

Listing 2: 计算关键变量统计均值与 CO_2 施肥效应的 R 脚本

```

1
2 # 检查并加载必要的包
3
4 if (!require(dplyr)) install.packages("dplyr")
5 library(dplyr)
6
7 # 定义基础路径
8 sim_dir <- "simulations"
9
10 analyze_scenario <- function(scenario_name, file_aCO2, file_eCO2) {
11
12     # 构建完整文件路径
13     path_a <- file.path(sim_dir, file_aCO2)
14     path_e <- file.path(sim_dir, file_eCO2)
15 }
```

```

16 # 检查文件是否存在
17 if (!file.exists(path_a) || !file.exists(path_e)){      warning(paste("文件未找到，跳过实验：" ,
18   scenario_name))
19   return(NULL)
20 }
21 df_a <- read.csv(path_a, header = TRUE, skip = 1)
22 df_e <- read.csv(path_e, header = TRUE, skip = 1)
23
24 # 筛选 2012-2016 年的数据
25 # 统一列名转换为小写，防止大小写差异 (GDAY输出通常是大写的 YEAR)
26 colnames(df_a) <- tolower(colnames(df_a))
27 colnames(df_e) <- tolower(colnames(df_e))
28
29 target_years <- 2012:2016
30
31 df_a_sub <- df_a %>% filter(year %in% target_years)
32 df_e_sub <- df_e %>% filter(year %in% target_years)
33
34 # 自动匹配变量名 (CW 和 Stem 的问题)
35 # 如果 CSV 里是 cw，将其复制为 stem 以便统一计算
36 if("cw" %in% colnames(df_a_sub)) {
37   df_a_sub$stem <- df_a_sub$cw
38   df_e_sub$stem <- df_e_sub$cw
39 }
40
41 # 定义计算均值的辅助函数
42 calc_mean_annual <- function(data, var_name, type="sum") {
43   if (!var_name %in% colnames(data)) {
44     warning(paste("变量未找到：" , var_name))
45     return(NA)
46   }
47
48   if (type == "sum") {
49     # 通量 (Flux): 先算每年的总和，再算几年的平均
50     # 例如 GPP (g/m2/day) -> 年总和 -> 年平均
51     annual_sums <- data %>% group_by(year) %>% summarize(val = sum(!is.na(sym(var_name)), na .
52       rm=TRUE))
53     return(mean(annual_sums$val))
54   } else {
55     # 库 (Pool): 状态变量，直接求这一段时间的平均值
56     # 例如 LAI, Stem Biomass
57     return(mean(data[[var_name]], na.rm=TRUE))
58   }
}

```

```

59
60 # 1. GPP (总初级生产力)- 通量 gpp_a <- calc_mean_annual(df_a_sub, "gpp", "sum")
61 gpp_e <- calc_mean_annual(df_e_sub, "gpp", "sum")
62
63 # 2. LAI (叶面积指数) - 状态量
64 lai_a <- calc_mean_annual(df_a_sub, "lai", "mean")
65 lai_e <- calc_mean_annual(df_e_sub, "lai", "mean")
66
67 # 3. Stem (茎干生物量) - 状态量
68 stem_a <- calc_mean_annual(df_a_sub, "stem", "mean")
69 stem_e <- calc_mean_annual(df_e_sub, "stem", "mean")
70
71 # 4. ET (蒸散发) - 通量
72 et_a <- calc_mean_annual(df_a_sub, "et", "sum")
73 et_e <- calc_mean_annual(df_e_sub, "et", "sum")
74
75 # 构建结果数据框
76 res <- data.frame(
77   Experiment = rep(scenario_name, 4),
78   Variable = c("GPP", "LAI", "Stem", "ET"),
79   Unit = c("g C m-2 yr-1", "m2 m-2", "g C m-2", "mm yr-1"),
80   aCO2_Mean = round(c(gpp_a, lai_a, stem_a, et_a), 2),
81   eCO2_Mean = round(c(gpp_e, lai_e, stem_e, et_e), 2)
82 )
83
84 # 计算 CO2 施肥效应 (%)
85 res$Fertilization_Effect_Pct <- round((res$eCO2_Mean - res$aCO2_Mean) / res$aCO2_Mean *
86   100, 2)
87
88 return(res)
89 }
90
91 # 实验一：标准情景
92 print("正在计算实验一...")
93 exp1 <- analyze_scenario("Exp 1: Standard", "simulation_aCO2.csv", "simulation_eCO2.csv")
94
95 # 实验二：降水减半 (Rev Rain)
96 print("正在计算实验二...")
97 exp2 <- analyze_scenario("Exp 2: Reduced Rain", "simulation_aCO2_rev_rain.csv", "simulation_eCO2_rev_rain.csv")
98
99 # 实验三：固定 C:N (Fix Leaf NC)
100 print("正在计算实验三...")
101 exp3 <- analyze_scenario("Exp 3: Fixed C:N", "simulation_aCO2_fixleafnc.csv", "simulation_eCO2_"

```

```

    fixleafnc.csv")
102 #合并结果final_table <- rbind(exp1, exp2, exp3)
103
104 # 导出 CSV (建议直接打开这个文件复制数据)
105 write.csv(final_table, "Experiment_Results_Calculated.csv", row.names = FALSE)

```

Listing 3: 验证叶片 N:C 比率的 R 脚本

```

1 # 读取实验一数据
2 df_a <- read.csv("simulations/simulation_aCO2.csv", header = TRUE, skip = 1)
3
4 # 计算叶片氮碳比 (shootnc)
5 # 注意：我们要避开 shoot 为 0 的情况（虽然在生长季一般不为0）
6 df_a$shootnc_calc <- ifelse(df_a$shoot > 0, df_a$shootn / df_a$shoot, NA)
7
8 # 筛选 2012-2016 年的数据
9 df_sub <- subset(df_a, year >= 2012 & year <= 2016)
10
11 # 统计分析
12 cat("== 实验一叶片氮碳比 (Shoot N:C) 统计 ==\n")
13 cat("设定下边界 (nf_min): 0.005\n")
14 cat("实际最小值:", min(df_sub$shootnc_calc, na.rm=TRUE), "\n")
15 cat("实际平均值:", mean(df_sub$shootnc_calc, na.rm=TRUE), "\n")
16 cat("实际最大值:", max(df_sub$shootnc_calc, na.rm=TRUE), "\n")
17
18 # 计算对应的 C:N 比
19 cat("\n== 对应的 C:N 比 ==\n")
20 cat("理论边界 C:N (1/0.005):", 1/0.005, "\n")
21 cat("实际运行 C:N:", 1/mean(df_sub$shootnc_calc, na.rm=TRUE), "\n")
22
23 # 绘图检查是否是一条直线
24
25 plot(df_sub$shootnc_calc, type='l', main="Shoot N:C Ratio (2012-2016)",
26       ylab="N:C Ratio", xlab="Days", ylim=c(0.004, 0.006))
27 abline(h=0.005, col="red", lty=2)

```