

祝介东北京力高泰科技有限公司

***R* 语言分析 *LI-6400* 和 *LI-6800* 光合仪的数据**



目录

表格	vii
插图	ix
前言	i
版权声明	iii
关于作者	v
1 R 软件与 Rstudio	1
1.1 R 软件	1
1.2 Rstudio	1
2 CO₂ 响应曲线的拟合	3
2.1 FvCB 模型	3
2.2 plantecophys 软件包	6
2.3 LI-6400XT CO ₂ 响应曲线的拟合	6
2.3.1 fitaci 函数介绍	6
2.4 使用 plantecophys 拟合 LI-6400XT CO ₂ 响应曲线数据	9
2.4.1 数据的前处理	9
2.4.2 使用示例	9
	iii

2.4.3	使用 ‘onepoint’ 单独计算 V_{cmax} 和 J_{max}	17
2.4.4	多条 CO_2 响应曲线的拟合	19
2.4.5	findCiTransition 函数	24
2.5	C4 植物光合	25
2.5.1	C4 植物光合速率的计算	26
3	气孔导度模型的拟合	27
3.1	BallBerry 模型	27
3.2	BBLeuning 模型	28
3.3	BBOptiFull 模型	28
3.4	fitBB 函数	28
3.5	fitBBs 函数	30
4	光合最优气孔导度耦合模型	33
4.1	FARAO 函数	33
5	光合气孔导度耦合模型	35
5.1	Photosyn 函数	36
5.1.1	Photosyn 使用举例	36
5.2	PhotosynEB 函数	41
5.3	PhotosynTuzet 函数	41
5.3.1	PhotosynTuzet 的参数	41
6	RHtoVPD 函数	43

<i>Contents</i>	v
7 光响应曲线的拟合	45
7.1 直角双曲线模型	45
7.1.1 直角双曲线模型的实现	46
7.2 非直角双曲线模型	51
7.2.1 非直角双曲线模型的实现	51
7.3 指数模型	55
7.3.1 指数模型的实现	55
7.4 直角双曲线的修正模型	59
7.4.1 直角双曲线修正模型的实现	59
8 关于非线性拟合的初始值	63
8.1 nlsLM 解决方案	64
8.2 作图比对法	65
8.2.1 实现过程	66
8.2.2 直观展示	69
8.3 自动多次尝试法	72
8.4 小结	73
9 LI-6800 的数据分析	75
9.1 数据格式	75
9.2 LI-6800 与 LI-6400 使用时的差别	75
9.3 光响应曲线注意事项	76
9.4 LI-6800 RACiR™ 的测量与拟合	76
9.5 RACiR™ 分析的实现	78
9.6 LI-6800 荧光数据分析	82
9.6.1 jip test 的实现	82

9.6.2	jip_test 软件包安装	83
9.6.3	read_files 及 read_dcfiles 函数	83
9.6.4	jip_test 及 jip_dctest 函数	84
9.6.5	jip_plot 及 jip_dcplot 函数	87
10	环境与配置	89

表格

2.1 推荐 LI-6400 整理后数据样式 9

2.2 onepoint 使用的数据 18

2.3 onepoint 法计算的结果 18

7.1 直角双曲线计算参数 49

7.2 非直角双曲线计算参数 54

7.3 指数模型计算参数 58

7.4 直角双曲线修正模型计算参数 61

9.1 推荐 LI-6800 整理后数据样式 75

9.2 jiptest 批量导入数据后的样式 83

9.3 jiptest DC 数据批量导入数据后的样式 84

9.4 jiptest 输出的计算参数 84

9.5 jiptest DC 数据输出的计算参数 86



插图

1.1 Rstudio 界面及功能	2
2.1 光合速率的不同的限制阶段	4
2.2 光合速率的不同的限制阶段	16
2.3 fitacis 作图结果	23
2.4 fitacis 作图结果	24
2.5 C4 植物 A-Ci 作图	26
5.1 VPD VS. An	38
5.2 PPFD VS. GS	38
5.3 PPFD VS. GS	39
5.4 supply VS. demand	40
7.1 直角双曲线模型拟合	49
7.2 非直角双曲线模型拟合	54
7.3 指数模型拟合	57
7.4 直角双曲线修正模型拟合	62
8.1 初步判断 alpha 的初始值	67
8.2 初修正后断 alpha 的初始值	68
8.3 检验作图法的初始值判断	70

8.4 多个 α 取值的差异	71
8.5 两种方法结果的对比展示	74
9.1 RACiR 方法与常规结果的比较	82
9.2 调制式测量的 ojip 曲线的快速预览	88
9.3 连续式测量的 ojip 曲线的快速预览	88

前言

本文内容均来自个人对相关材料的理解，并未经过权威认证，如有异议，请参考 **R** 软件及相关软件包的使用手册或相关模型的文献。本文所使用代码均只做示例代码使用，在做实际分析时，请结合自己的实验数据及模型使用条件做相应的调整。本文内容仅针对 LI-6800 与 LI-6400XT 光合仪数据，尤其是 RACiR™ 部分，仅针对 LI-6800 进行分析，且目前市面上仅 LI-6800 光合仪可完成此项数据的测量。如您有 LI-6400XT 或 LI-6800 使用操作相关方面的问题请发送至：

zhujiedong@ecotek.com.cn

如有与本文数据分析相关的内容请发送至我个人邮箱（本文是业余爱好，内容虽与公司仪器相关，但并不是我领薪水的工作内容）：

zhujiedong@yeah.net

以便我们对相关内容做出修正。

当然，如果您有 github 帐号，优先欢迎在 github 提交：

github 地址¹

如有其他仪器问题或者仪器购买需求，请使用下面方式与我们联系：

- 北京力高泰科技有限公司
- 网址：<http://www.ecotek.com.cn>
- 电话：010-51665551
- 电子邮件：info@ecotek.com.cn
- 地址：北京市西城区西直门南大街 2 号成铭大厦 A 座 22F

¹<https://github.com/zhujiedong/photoanalysis>



版权声明

本文旨在对广大 **LI-6400XT** 和 **LI-6800** 光合仪的用户提供一个数据分析的参考，本人并未从中获取任何利益，内容错误疏漏之处，欢迎指正。本人保留一切权利，禁止一切将本文内容用于商业用途的行为，禁止其他仪器公司抄袭及使用。



关于作者

祝介东，北京力高泰科技有限公司² 工程师，本文内容为我在售后服务时所住的酒店所作（遍布国内大多数省份），本为我打发出差住酒店漫漫长夜的业余爱好，此部分内容并非公司对我的职位所要求的内容，也并非我公司提供的服务内容，因此属于个人作品，仅供参考，我所在公司对此文内容不负任何责任。文中代码得到了公司技术部同事：刘美玲、徐粒、焦阳、王昭、张启尧、张云飞等的测试，其中焦阳贡献了 `jipetest` 包中基础荧光参数计算的代码，在此一并感谢。

文中内容如对您有帮助，我很欣慰。但本人水平有限，错误与疏漏之处还请谅解，欢迎讨论，欢迎提出宝贵意见及建议。本人或单位其他售后人员仅有提供相关仪器操作或故障解决的义务，处理数据和分析数据并非我们的本职工作，如有相关需还请多参考本文内容或网上相关资料，不足之处还请谅解。

²<http://www.ecotek.com.cn/>



1

R 软件与 *Rstudio*

1.1 R 软件

R 语言的官方网站是 R¹，与 R 语言有关的网站还有 CRAN（镜像），其主站网址是：CRAN²。

1.2 Rstudio

尽管 R 的功能十分强大，但是其作为一个命令行工具³，在实际使用中尤为不便，因此，一款集成的开发环境十分有必要。Rstudio⁴是一款专门针对 R 开发的一个集成环境，同时也支持其他多种语言，用户界面十分友好，支持代码高亮，拼写提示，作图展示等功能，因此我们推荐使用 Rstudio 对我们的数据进行分析。

R 软件和 Rstudio 的安装十分方便，我们按照各自系统的安装方式安装即可，安装后界面及各区功能如下：

本文的主要内容并非介绍这些软件的功能，因此，关于软件的操作部分请仔细阅读相关资料，网络上有大量的免费资源及教程，有需要的同学可自行搜索。

¹<http://www.r-project.org>

²<http://www.cran.r-project.org>

³实际上在 Windows 系统下安装的时候提供一个十分简陋的 GUI。

⁴<https://www.rstudio.com/products/RStudio/>

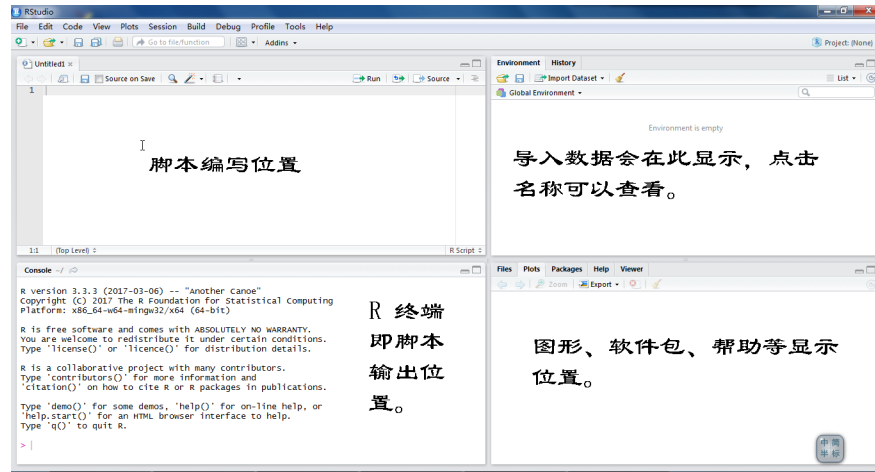


图 1.1: Rstudio 界面及功能

注：R 与 Rstudio 均非我公司产品，而且均免费或者有免费版本，因此请勿邮件或电话索要此两款软件。

2

CO_2 响应曲线的拟合

2.1 FvCB 模型

在 `plantecophys` 包中使用的模型为 [Farquhar et al. \(1980\)](#) 建立的 C3 植物模型 FvCB, 其基于 C3 植物碳反应的三个阶段:

- 核酮糖-1,5-双磷酸羧化酶/加氧酶 (Rubisco) 的催化下, 核酮糖-1,5-双磷酸 (RuBP) 与 CO_2 发生羧化作用, 生成 3-磷酸甘油酸 (PGA)。
- 在腺苷三磷酸 (ATP) 和还原型烟酰胺腺嘌呤二核苷酸磷酸 (NADPH) 的作用下, PGA 被还原成磷酸丙糖 (TP)。每 6 个 TP 中有 1 个输出到细胞液中, 用于蔗糖或者淀粉的合成。
- 剩下的 5 个 TP 在 ATP 的作用下再生为 3 个 RuBP。一部分再生的 RuBP 在 Rubisco 的催化下被氧化成 PGA 和 2-磷酸乙醇酸, 2-磷酸乙醇酸在 ATP 的作用下形成 PGA, 并且释放 CO_2 (光呼吸)。

在光照下, C3 植物净光合速率 (A) 取决于 3 个同时存在的速率: RuBP 羧化速率 (V_c)、RuBP 氧化速率 (或光呼吸速率, V_o) 和线粒体在光照下的呼吸速率 (或明呼吸速率, R_d ; 此名为了与暗呼吸速率对应和区分)。RuBP 氧化过程中每结合 1 mol O_2 就会释放 0.5 mol CO_2 。因此, 净光合速率 A 的计算为:

$$A = V_c - 0.5V_o - R_d \quad (2.1)$$

线粒体 R_d 不同于暗呼吸速率 (R_n)。 R_n 是叶片在黑暗中的线粒体呼吸速

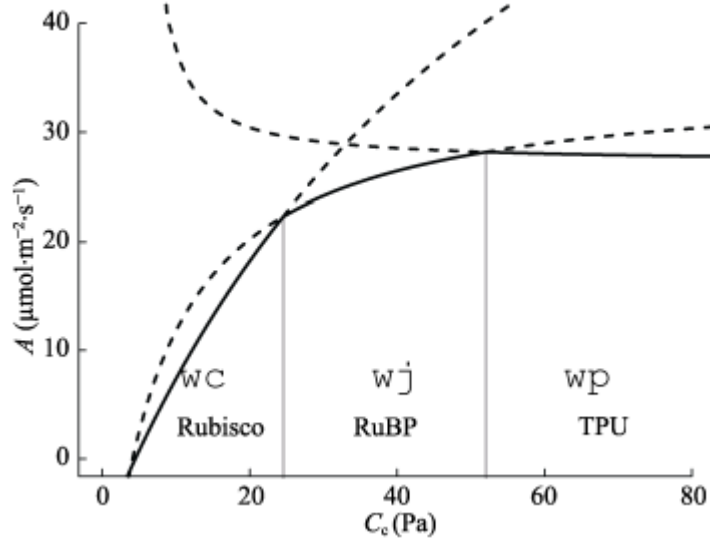


图 2.1: 光合速率的不同的限制阶段

率, 随着光照的增加, 线粒体呼吸速率下降。因此 $R_d < R_n$ 在黑暗条件下测定的叶片 CO_2 交换速率即 R_n , 但是 R_d 的测定比较困难, 因为光照条件下 R_d 与 V_c 、 V_o 同时存在。Hikosaka et al. (2006) 总结了几种测定 R_d 的方法, 式 (2.1) 又可表达为:

$$A = V_c (1 - 0.5\alpha) R_d \quad (2.2)$$

式 (2.2) 中 α 为氧化速率和羧化速率的比值, 由 Rubisco 动力学常数确定:

$$\alpha = \frac{V_o}{V_c} = \frac{O}{C_c} \times \frac{V_{omax} K_c}{V_{cmax} K_o} = \frac{O}{C_c} \times \frac{1}{S_o} \quad (2.3)$$

式 (2.3) 中, C_c 和 O 分别为叶绿体部位 CO_2 和 O_2 浓度。 C_c 和 O 通常以气体摩尔分数 ($\mu mol \cdot mol^{-1}$) 或分压 (Pa) 表示, 但光合过程是在叶绿体的液相基质中发生的, 用分压表示更加恰当。 K_c 与 K_o 为 Rubisco 羧化 (氧化) 的米氏常数, 代表了羧化 (氧化) 速率达到最大羧化 (氧化) 速率一半时的 CO_2 和 O_2 浓度。是 Rubisco 特异性因子, S_o 表示 Rubisco 对 CO_2 和 O_2 的偏好程度。

当 $A = R_d$, 即 RuBP 羧化的 CO_2 吸收速率刚好等于 RuBP 氧化的 CO_2 释放速率 ($V_c = 2V_o$) 时, $\alpha = 0.5$ 。此时叶绿体的 CO_2 浓度就是叶绿体 CO_2 光合补偿点, 标记为 Γ^* 。即:

$$\Gamma^* = \frac{0.5O}{S_o} \quad (2.4)$$

由公式 (2.3)和公式 (2.4)可得:

$$\alpha = \frac{2\Gamma^*}{C_c} \quad (2.5)$$

代入公式 (2.2) 得到:

$$A = V_c \left(1 - \frac{\Gamma^*}{C_c}\right) R_d \quad (2.6)$$

在 C_c 浓度很低的时候, RuBP 供应充足 (图 2.1 Rubisco 阶段), V_c 等于 Rubisco 所能支持的羧化速率 w_c , :

$$w_c = \frac{V_{cmax} C_c}{C_c + K_c \left(1 + \frac{O}{K_o}\right)} \quad (2.7)$$

随着 C_c 浓度的增加, Rubisco 支持的羧化速率超过了 RuBP 供应速率, V_c 受 RuBP 再生速率的限制 (图 2.1 RuBP 阶段), 此时 V_c 由 RuBP 的再生速率限制, 而 RuBP 又由电子传递速率 (J) 决定, 故:

$$w_j = \frac{J C_c}{4C_c + 8\Gamma^*} \quad (2.8)$$

当 C_c 浓度很高, 光合磷酸化超过了淀粉和蔗糖的合成速率的时候, V_c 受到 TP 利用速率 (V_p) 的限制 (图 2.1 TPU 阶段), 一般情况下,

$$w_p = \frac{3V_p C_c}{C_c - \Gamma^*} \quad (2.9)$$

最终, C3 植物叶片的光合速率 A 由 w_c 、 w_j 、 w_p 的最小者决定 (图 2.1 实现部分), 当 $c > \Gamma^*$ 时:

$$A = \min\{w_c, w_j, w_p\} \left(1 - \frac{\Gamma^*}{C_c}\right) - R_d \quad (2.10)$$

2.2 plantecophys 软件包

Remko A. Duursma 在 2015 年发表了一篇文章 [Duursma \(2015\)](#), `plantecophys` 是其开发的一个 R 包工具集, 用于对叶片气体交换数据进行分析 and 建模。实现了如下功能:

- CO_2 响应曲线 (A-Ci curves) 的拟合、作图及模拟。
- 不同气孔导度模型。
- 根据 Cowan-Farquhar 的假设估算最适的气孔导度。
- 耦合气体交换模型的实现。
- 基于 Ci 模拟 C4 光合。
- RHtoVPD: 常用单位的转换 (相对湿度、水汽压亏缺、露点温度)。

各参数的基本用法请参考后文内容, 或官方帮助文档:

`plantecophys`¹。

2.3 LI-6400XT CO_2 响应曲线的拟合

LI-6400XT CO_2 响应曲线的拟合需要借助 `plantecophys` 的 `fitaci` 实现, `fitaci` 函数为根据 FvCB 模型对 A-Ci 曲线的测量数据进行拟合, 并估算 J_{max} 、 V_{cmax} 、 R_d 及他们的标准差, 并根据 [Medlyn et al. \(2002\)](#) 的方法考虑了温度的影响。

2.3.1 `fitaci` 函数介绍

`fitaci` 的用法如下²:

¹<https://cran.r-project.org/web/packages/plantecophys/plantecophys.pdf>

²仅针对 C3 植物

```

fitaci(data, varnames = list(ALEAF = "Photo",
  Tleaf = "Tleaf", Ci = "Ci", PPFD = "PARi",
  Rd = "Rd"), Tcorrect = TRUE, Patm = 100,
  citransition = NULL, quiet = FALSE,
  startValgrid = TRUE, fitmethod =
  c("default", "bilinear", "onepoint"),
  algorithm = "default", fitTPU = FALSE,
  useRd = FALSE, PPFD = NULL, Tleaf = NULL,
  alpha = 0.24, theta = 0.85, gmeso = NULL,
  EaV = 82620.87, EdVC = 0, delsC = 645.1013,
  EaJ = 39676.89, EdVJ = 2e+05,
  delsJ = 641.3615, GammaStar = NULL,
  Km = NULL, id = NULL, ...)

## S3 method for class 'acifit'
plot(x, what = c("data", "model", "none"),
      xlim = NULL, ylim = NULL, whichA = c(
        "Ac", "Aj", "Amin", "Ap"), add = FALSE,
pch = 19, addzeroline = TRUE, addlegend =
  !add, legendbty = "o", transitionpoint = TRUE,
linecols = c("black", "blue", "red"), lwd = c(1,
2), ...)

```

主要参数注释:

- data: 需要分析的数据, 必须为 data.frame³ 格式。
- varnames: 数据的表头, 此处函数默认的表头为 LI-6400 的表头, 分析 LI-6400 的数据时可以不用填写, 直接使用默认的参数即可⁴。
- Tcorrect: 如果为 TRUE, 那么 J_{max} 、 V_{cmax} 的结果为温度校正结果, 若 Tcorrect = FALSE, 则为测量温度下的结果。

³具体参考 R 语言相关文档, 为 R 语言最常用的数据格式

⁴在 R 中, 使用参数的值为默认值时不用填写该参数, 例如使用默认选项分析 LI-6400 数据时, 可只填写 data 项, 具体参考 R 的相关入门手册

- Patm: 为外界大气压。
- citransition: 参见详, 若提供该选项, 则 J_{max} 、 V_{cmax} 的区域则分别拟合⁵。
- fitmethod: 参见详解。
- fitTPU: 是否拟合 TPU 限制, 默认为 FALSE, 参见详解。
- x: 对于 plot.acifit, x 为 fitaci 返回的对象, 简单理解为将 fitaci 函数拟合结果赋值给一个变量, 此处 plot 函数实际上为 plot.acifit。
- what: 利用基础做图工具, 默认为对数据和模型进行作图。
- whichA: 默认为对所有的光合速率进行作图 ($A_j=J_{max}$ -limited (蓝色), $A_c=V_{cmax}$ - limited (红色), Hyperbolic minimum (黑色)), TPU-limited rate (A_p , 如果模型有计算结果)。

其他参数请参考 FvCB 模型 [Farquhar et al. \(1980\)](#) 或查看 plantecophys 的帮助文档。

2.3.1.1 fitaci 函数详解

- 默认为非线性拟合, 详见 [Duursma \(2015\)](#)。
- bilinear 方法使用两次线性拟合方法首先拟合 V_{cmax} 和 R_d , 然后在拟合 J_{max} , 过渡点的选择为对所有数据拟合最适的点, 类似 [Gu et al. \(2010\)](#) 的方法。该方法的优势时无论如何, 都会返回拟合结果, 尤其是非线性拟合失败时使用该方法, 但若默认方法失败时, 需首先检查是否数据存在问题。两种拟合方法的结果有轻微的差别⁶。
- onepoint 参考 [De Kauwe et al. \(2016\)](#)。
- citransition 使用时, 数据将被区分为 V_{cmax} 限制 ($C_i < citransition$) 区域, 以及 J_{max} 限制 ($C_i > citransition$) 区域。
- fitTPU: 如果要计算 TPU, 要设置 fitTPU = TRUE, 并且 fittingmethod = "bilinear"。但需要注意, 当 TPU 被计算时, 没有 J_{max} 限制的点的存在是可能的。TPU 限制的发生是在 A 值不随 CO_2 的增加而增加时发生的, 因此计算 TPU 是否有返回值, 取决于测量数据是否有此情况出现。

⁵为 Rubisco 和 RuBP 限制的 C_i 转换点, 物种间差异较大, 可以通过预实验确定

⁶若默认拟合方法失败, 数据也无问题, 那么是非线性拟合初始值设定的原因

2.4 使用 *plantecophys* 拟合 LI-6400XT CO₂ 响应曲线数据

2.4.1 数据的前处理

虽然 R 软件支持直接导入 *xlsx* 的数据，但因为 LI-6400XT 的数据记录文件内有其他空行或 *remark* 等内容，增加了处理代码的量，故而推荐将其数据先整理为如表 2.1 样式，并另存为 *csv* 格式⁷：

表 2.1: 推荐 LI-6400 整理后数据样式

Obs	HHMMSS	FTime	EBal.	Photo	Cond	Ci	Trmmol
1	15:46:59	271.5	0	14.2848912	0.2730691	286.39751	2.226126
2	15:48:26	358.0	0	10.6562220	0.2826303	217.32002	2.292845
3	15:49:54	446.0	0	6.4525814	0.2909460	150.67623	2.361704
4	15:51:26	538.5	0	1.7971569	0.3057164	85.82530	2.459459
5	15:52:54	626.5	0	-0.6575974	0.3150002	53.47985	2.515992
6	15:54:50	742.5	0	15.4296572	0.3255415	292.56161	2.579840

2.4.2 使用示例

plantecophys 并非 *base* 的安装包，首次使用需要从 *CRAN* 安装，可以使用图形界面安装，也可以直接用命令行安装⁸，推荐同时安装依赖。

```
install.packages("plantecophys", dependencies = TRUE)
```

⁷即仅保留测量值，删除其他所有头文件、空行、*remark* 等信息

⁸首次使用安装，更换电脑或者升级 R 软件后，如果没有拷贝 *library*，也需要运行安装命令

```
# 载入 plantecophys
library("plantecophys")

# 利用 read.csv 读取数据文件，
# 我的路径为当前工作路径的 data 文件夹内
aci <- read.csv("./data/aci.csv")

# 防止可能出现的 NA 值
aci <- subset(aci, Obs > 0)

# 不修改默认参数对数据进行拟合
acifit <- fitaci(aci)
# 查看拟合结果的参数名称，方便导出数据使用
attributes(acifit)
```

```
## $names
## [1] "df"           "pars"         "nlsfit"       "Tcorrect"
## [5] "Photosyn"     "Ci_transition" "Ci_transition2" "Rd_measured"
## [9] "GammaStar"    "Km"           "kminput"      "gstarinput"
## [13] "fitmethod"    "citransition" "gmeso"        "fitTPU"
## [17] "alphag"       "RMSE"         "runorder"
##
## $class
## [1] "acifit"
```

```
# 查看拟合结果
summary(acifit)
```

```
## Result of fitaci.
##
## Data and predictions:
```

```

##          Ci      Ameas      Amodel      Ac      Aj      Ap      Rd VPD
## 5      53.47985 -0.6575974 -0.5146882 -0.3552036  0.000000 1000 0.159449 1.5
## 4      85.82530  1.7971569  1.9292621  2.0888575  5.068534 1000 0.159449 1.5
## 3     150.67623  6.4525814  6.4176037  6.5777528 12.755502 1000 0.159449 1.5
## 2     217.32002 10.6562220 10.5354626 10.6965875 17.519644 1000 0.159449 1.5
## 1     286.39751 14.2848912 14.3365887 14.4993980 20.749310 1000 0.159449 1.5
## 6     292.56161 15.4296572 14.9749157 15.1383702 20.852616 1000 0.159449 1.5
## 7     292.96456 15.7134791 15.0564801 15.2200522 20.831098 1000 0.159449 1.5
## 8     450.64285 22.2659015 23.0115187 23.1975997 25.186939 3000 0.159449 1.5
## 9     622.03873 26.5135040 27.6485003 30.4281393 27.837462 3000 0.159449 1.5
## 10    992.08737 30.3898585 30.6300461 42.3998173 30.797660 3000 0.159449 1.5
## 11   1558.96968 33.6267056 32.6638021 54.9948264 32.828110 3000 0.159449 1.5
## 12   1756.16396 33.3152783 33.0981490 58.5844507 33.261965 3000 0.159449 1.5
##          Tleaf      Cc      PPFD Patm Ci_original
## 5      31.12332      53.47934 1800.490   100      53.47985
## 4      30.99093      85.82723 1800.558   100      85.82530
## 3      30.82872     150.68265 1800.140   100     150.67623
## 2      30.63983     217.33057 1800.524   100     217.32002
## 1      30.46890     286.41186 1800.701   100     286.39751
## 6      31.26338     292.57660 1799.923   100     292.56161
## 7      31.41866     292.97963 1799.975   100     292.96456
## 8      31.54122     450.66588 1799.826   100     450.64285
## 9      31.63493     622.06640 1799.578   100     622.03873
## 10     31.73910     992.11803 1800.055   100     992.08737
## 11     31.86938    1559.00238 1800.022   100    1558.96968
## 12     31.96654    1756.19709 1799.585   100    1756.16396
##
## Root mean squared error:  1.889701
##
## Estimated parameters:
##          Estimate Std. Error
## Vcmax    49.261616  1.5152405
## Jmax    126.620537  2.2816267

```

```
## Rd      0.159449  0.4001302
## Note: Vcmax, Jmax are at 25C, Rd is at measurement T.
##
## Curve was fit using method:  default
##
## Parameter settings:
## Patm = 100
## alpha = 0.24
## theta = 0.85
## EaV = 82620.87
## EdVC = 0
## delsC = 645.1013
## EaJ = 39676.89
## EdVJ = 2e+05
## delsJ = 641.3615
##
## Estimated from Tleaf (shown at mean Tleaf):
## GammaStar = 58.61138
## Km = 1223.279
```

```
acifit_linear <- fitaci(aci, fitmethod = "bilinear")
summary(acifit_linear)
```

```
## Result of fitaci.
##
## Data and predictions:
```

##	Ci	Ameas	Amodel	Ac	Aj	Ap	Rd
## 5	53.47985	-0.6575974	-0.7389447	-0.3560483	0.000000	1000	0.3828608
## 4	85.82530	1.7971569	1.7108198	2.0938246	5.138366	1000	0.3828608
## 3	150.67623	6.4525814	6.2098476	6.5933940	12.931333	1000	0.3828608
## 2	217.32002	10.6562220	10.3375299	10.7220229	17.761317	1000	0.3828608
## 1	286.39751	14.2848912	14.1477696	14.5338762	21.035722	1000	0.3828608

```
## 6 292.56161 15.4296572 14.7876514 15.1743678 21.139657 1000 0.3828608
## 7 292.96456 15.7134791 14.8694171 15.2562440 21.117718 1000 0.3828608
## 8 450.64285 22.2659015 22.8464806 23.2527613 25.533374 3000 0.3828608
## 9 622.03873 26.5135040 27.8030690 30.5004944 28.220254 3000 0.3828608
## 10 992.08737 30.3898585 30.8295808 42.5006398 31.221072 3000 0.3828608
## 11 1558.96968 33.6267056 32.8913778 55.1255986 33.279305 3000 0.3828608
## 12 1756.16396 33.3152783 33.3316070 58.7237587 33.719013 3000 0.3828608
##      VPD      Tleaf      Cc      PPFD Patm Ci_original
## 5 1.5 31.12332 53.47911 1800.490 100 53.47985
## 4 1.5 30.99093 85.82701 1800.558 100 85.82530
## 3 1.5 30.82872 150.68244 1800.140 100 150.67623
## 2 1.5 30.63983 217.33037 1800.524 100 217.32002
## 1 1.5 30.46890 286.41167 1800.701 100 286.39751
## 6 1.5 31.26338 292.57641 1799.923 100 292.56161
## 7 1.5 31.41866 292.97944 1799.975 100 292.96456
## 8 1.5 31.54122 450.66572 1799.826 100 450.64285
## 9 1.5 31.63493 622.06656 1799.578 100 622.03873
## 10 1.5 31.73910 992.11823 1800.055 100 992.08737
## 11 1.5 31.86938 1559.00260 1800.022 100 1558.96968
## 12 1.5 31.96654 1756.19733 1799.585 100 1756.16396
##
## Root mean squared error: 2.013045
##
## Estimated parameters:
##      Estimate Std. Error
## Vcmax 49.3787547 3.4815555
## Jmax 128.5546403 NA
## Rd 0.3828608 0.4697008
## Note: Vcmax, Jmax are at 25C, Rd is at measurement T.
##
## Curve was fit using method: bilinear
##
## Parameter settings:
```

```
## Patm = 100
## alpha = 0.24
## theta = 0.85
## EaV = 82620.87
## EdVC = 0
## delsC = 645.1013
## EaJ = 39676.89
## EdVJ = 2e+05
## delsJ = 641.3615
##
## Estimated from Tleaf (shown at mean Tleaf):
## GammaStar = 58.61138
## Km = 1223.279
```

```
# 仅查看拟合参数，比较两种拟合参数的差异
coef(acifit_linear)
```

```
##      Vcmax      Jmax      Rd
## 49.3787547 128.5546403 0.3828608
```

```
coef(acifit)
```

```
##      Vcmax      Jmax      Rd
## 49.261616 126.620537 0.159449
```

```
# 设置作图参数，图形的边距及分为 1 行两列输出图形
par(mar = c(4.5, 4.5, 2, 2))
par(mfrow = c(1, 2))
# 对两种拟合参数的结果作图，查看模型拟合是否正常
```

```

plot(acifit, addlegend = FALSE)
legend(x = 500, y = 10,
       legend = c(expression(paste(A[c])),
                     expression(paste(A[j])),
                     "Limiting rate"),
       lty = c(1, 1, 1),
       col = c("red", "blue", "black")
       )
mtext(" fitmethod = 'default' ")

plot(acifit_linear, addlegend = FALSE)
legend(x = 500, y = 10,
       legend = c(expression(paste(A[c])),
                     expression(paste(A[j])),
                     "Limiting rate"),
       lty = c(1, 1, 1),
       col = c("red", "blue", "black")
       )
mtext("fitmethod = 'bilinear' ")

```

如果需要导出数据做他用，直接根据 `attributes` 中看到的名称，选择对应的数据导出即可，如果使用 Rstudio 的话，其自动补全的功能在选择数据上更方便。例如导出预测值和系数分别使用如下方式：

```

# 将模型拟合结果中 df（即计算数据）赋给 predictaci,
# 并用 write.csv 导出
predictaci <- acifit$df
write.csv(acifit$df, file = "acipredict.csv")
write.csv(coef(acifit), file = "coefaci.csv")

```

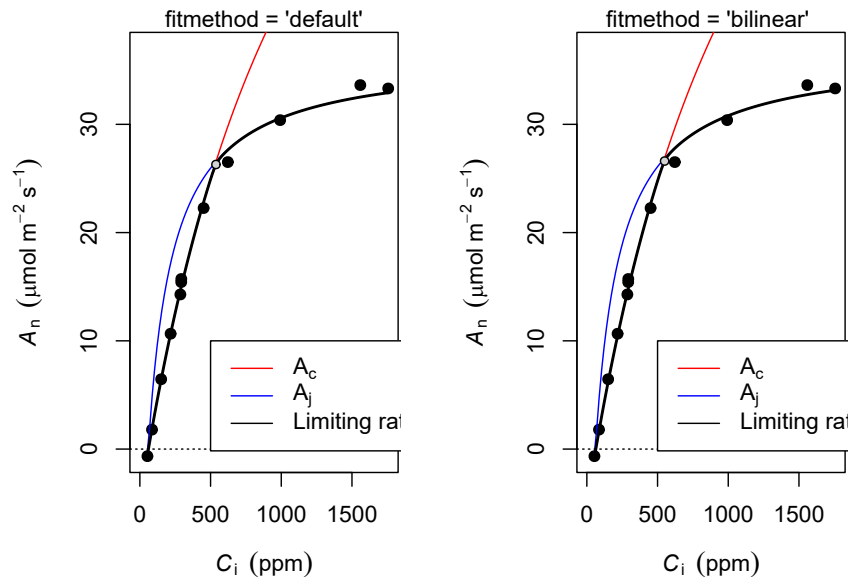


图 2.2: 光合速率的不同的限制阶段

需要注意的是，因为非线性拟合需要一个初始值，因此，使用默认方式（非线性拟合）的时候，会存在可能的拟合失败现象，此时可以使用 `fitmethod = "bilinear"`，二者结果略有差别。

2.4.2.1 `fitmethod = "onepoint"` 介绍

De Kauwe et al. (2016) 发表了关于 one point 方法计算 V_{cmax} 和 J_{max} 方法的文章，在 2017 年 11 月的更新中，plantecophys 增加了响应的 R 软件实现方法，该方法并非使用一个点计算 V_{cmax} 和 J_{max} ，而是对数据集中的每一个点的值进行估计，使用的方法为逆向了光合作用方程。输出为对每个原始数据加入了 V_{cmax} 和 J_{max} ，当然一如既往的可以使用温度校准的方法。并不建议该方法应用于整个 AC_i 曲线的数据，它的假设是在外部环境 CO_2 浓度和饱和光下，受到 Rubisco 所化速率的限制而不是 RUBP 的限制。

基于上面的描述，他们的模型如下：

$$\hat{V}_{cmax} = (A_{sat} + R_{day}) \frac{C_i + K_m}{C_i - \Gamma^*} \quad (2.11)$$

其中: K_m 为米氏常数, 其计算为:

$$K_m = K_c(1 + \frac{O_i}{K_o}) \quad (2.12)$$

未知参数均由文献中的方法进行计算, 具体可参考 De Kauwe et al. (2016) 的原文, 但上述方法的缺陷为还要使用 ACi 曲线来估算 R_{day} , 因此作者使用了 1.5% V_{cmax} 作为 R_{day} , 因此公式 (2.11) 可变换为:

$$\hat{V}_{cmax} = A_{sat}(\frac{C_i + K_m}{C_i - \Gamma^*} - 0.015) \quad (2.13)$$

另一个重要的模型的假设为 J_{max} 与 V_{cmax} 是成比例的, J_{max} 的计算是通过 C_i transition point 来实现的, 文章中的比值均值为 1.9, 范围在 1.68 ~ 2.14 之间。

2.4.3 使用 ‘onepoint’ 单独计算 V_{cmax} 和 J_{max}

目前我手头没有相应数据, 仅有使用 LI-6400 测试 auto log 2 时的一个数据, 我们用这个来示范该方法的使用:

```
one_data <- read.csv("./data/onepoint.csv")
knitr::kable(head(one_data), booktabs = TRUE,
               caption = 'onepoint 使用的数据')
```

数据如上所示, 为同一个叶片连续记录数据, 故所有的光合速率十分接近。

使用方法:

```
library(plantecophys)

one_data <- subset(one_data, Obs > 0)
```

表 2.2: onepoint 使用的数据

Obs	HHMMSS	FTime	EBal.	Photo	Cond	Ci	Trmmol	VpdL	CTleaf
1	14:12:43	347.0	0	10.519216	0.1369336	263.0905	1.963046	1.445782	25.89411
2	14:13:04	369.0	0	10.361122	0.1357092	264.1135	1.946022	1.445559	25.89542
3	14:13:25	373.0	0	10.207166	0.1342114	264.8348	1.925734	1.445717	25.89660
4	14:13:41	389.5	0	9.547416	0.1281947	269.6337	1.854272	1.454248	25.94020
5	14:16:16	540.0	0	10.288968	0.1376602	267.9198	2.009217	1.471788	26.01813
6	14:16:32	555.5	0	10.178603	0.1381995	269.5898	2.016100	1.471138	26.01657

表 2.3: onepoint 法计算的结果

aci_fit.Photo	aci_fit.Vcmax	aci_fit.Jmax
10.519216	47.06893	71.73759
10.361122	46.22091	70.51059
10.207166	45.44617	69.35838
9.547416	41.86217	64.24400
10.288968	45.09335	69.36954
10.178603	44.37360	68.41628

```

one_data$Rd <- rep(-0.5, length(one_data$Obs))
aci_fit <- fitaci(one_data, fitmethod = "onepoint")

knitr::kable(head(data.frame(aci_fit$Photo,
                             aci_fit$Vcmax, aci_fit$Jmax)),
              booktabs = TRUE,
              caption = 'onepoint 法计算的结果')

```

需要注意，为保证结果的精确，如果不设定 Rd，也即文献中的 Rday，模型是无法计算的，因此上面的示例中虚构了一个，实际操作用一般使用标准的 ACi 测量计算。

2.4.4 多条 CO_2 响应曲线的拟合

`fitacis` 函数实际上是 `fitaci` 函数的扩展，方便一次拟合多条曲线⁹。函数的参数如下：

```
fitacis(data, group, fitmethod = c("default",
  "bilinear"), progressbar = TRUE,
  quiet = FALSE, id = NULL, ...)

## S3 method for class 'acifits'
plot(x, how = c("manyplots", "oneplot"),
  highlight = NULL, ylim = NULL,
  xlim = NULL, add = FALSE, what = c("model",
  "data", "none"), ...)
```

主要参数详解：

实际上 `fitacis` 与 `fitaci` 模型算法完全一致，只不过增加了一个 `group` 参数，用于区分不同测量的数据，具体请参考举例内容。

2.4.4.1 `fitacis` 函数应用举例

下文代码根据 *plantecophys* 中的示例代码修改，进行演示，原代码请参考其帮助文档。

```
library(plantecophys)
# 只提取前 10 个不同测量的数据，节省时间进行举例
manyacidat2 <- droplevels(manyacidat[manyacidat$Curve %in%
  levels(manyacidat$Curve)[1:10],])

# 对多条曲线进行拟合，使用 bilinear 方法，
```

⁹需要注意，此时 `fitmethod` 一般推荐使用 `bilinear`。

仅仅因为其比非线性拟合节省时间

```
fits <- fitacis(manyacidat2, group = "Curve", fitmethod="bilinear")
```

```
##
```

```
|
|                                     | 0%
|
|=====| 10%
|
|=====| 20%
|
|=====| 30%
|
|=====| 40%
|
|=====| 50%
|
|=====| 60%
|
|=====| 70%
|
|=====| 80%
|
|=====| 90%
|
|=====| 100%
```

拟合结果为 *list*, 我们可以只提取第一个的拟合结果

```
fits[[1]]
```

```
## Result of fitaci.
```

##

Data and predictions:

##		Ci	Ameas	Amodel	Ac	Aj	Ap	Rd
## 2	53.23129	-0.4401082	0.1014381	0.9601119	2.548123	1000	0.8586158	
## 3	79.47367	2.4824630	2.1937702	3.0526198	7.036734	1000	0.8586158	
## 4	116.74688	5.4531712	4.9419337	5.8011511	11.392394	1000	0.8586158	
## 5	188.00801	9.7099879	9.5705964	10.4310194	16.447715	1000	0.8586158	
## 6	278.44662	14.8225766	14.4261545	15.2897486	19.977583	1000	0.8586158	
## 7	343.03259	17.7982155	17.4602014	18.3289218	21.639847	1000	0.8586158	
## 1	344.72152	17.9244012	17.3165146	18.1849643	21.534276	1000	0.8586158	
## 14	344.74839	16.7933747	17.6853306	18.5545917	21.774261	1000	0.8586158	
## 8	588.08078	23.8925326	24.1309683	27.0327638	25.020148	3000	0.8586158	
## 9	833.25547	26.5674409	25.7783026	33.0856065	26.647921	3000	0.8586158	
## 10	1136.99222	25.9787890	26.8108335	38.1296944	27.676768	3000	0.8586158	
## 11	1436.86370	26.6110657	27.5409345	42.0628463	28.405453	3000	0.8586158	
## 12	1536.46772	27.4018784	27.7965881	43.3773689	28.660781	3000	0.8586158	
## 13	1731.76400	28.6752069	28.0952804	45.2475932	28.959041	3000	0.8586158	

##	VPD	Tleaf	Cc	PPFD	Patm	Ci_original
## 2	1.5	24.55873	53.23139	1799.959	100	53.23129
## 3	1.5	24.58292	79.47586	1799.590	100	79.47367
## 4	1.5	24.71278	116.75183	1799.819	100	116.74688
## 5	1.5	24.73687	188.01759	1800.371	100	188.00801
## 6	1.5	24.67508	278.46106	1800.233	100	278.44662
## 7	1.5	24.76596	343.05006	1799.575	100	343.03259
## 1	1.5	24.51593	344.73886	1800.356	100	344.72152
## 14	1.5	24.94098	344.76609	1799.964	100	344.74839
## 8	1.5	24.83785	588.10494	1799.477	100	588.08078
## 9	1.5	24.91185	833.28127	1799.969	100	833.25547
## 10	1.5	24.87314	1137.01906	1799.525	100	1136.99222
## 11	1.5	24.95914	1436.89126	1799.615	100	1436.86370
## 12	1.5	25.04542	1536.49554	1799.784	100	1536.46772
## 13	1.5	25.07566	1731.79212	1799.160	100	1731.76400

##

```
## Root mean squared error: 2.196037
##
## Estimated parameters:
##           Estimate Std. Error
## Vcmax  65.0009909  1.3720635
## Jmax   131.7980133           NA
## Rd      0.8586158  0.2876248
## Note: Vcmax, Jmax are at 25C, Rd is at measurement T.
##
## Curve was fit using method: bilinear
##
## Parameter settings:
## Patm = 100
## alpha = 0.24
## theta = 0.85
## EaV = 82620.87
## EdVC = 0
## delsC = 645.1013
## EaJ = 39676.89
## EdVJ = 2e+05
## delsJ = 641.3615
##
## Estimated from Tleaf (shown at mean Tleaf):
## GammaStar = 42.31453
## Km = 698.2084
```

```
# 所有参数均可传递到 plot.acifit, 例如:
# 将所有数据点作图, 并将数据点标记为蓝色
par(mar = c(4.5, 4.5, 2, 2))
par(mfrow = c(1, 2))
plot(fits, how="oneplot", what="data", col="blue", addlegend = FALSE)
# 在上图的基础上增加模型拟合的线
```

```
plot(fits, how="oneplot", what="data", col="blue", addlegend = FALSE)
plot(fits, how="oneplot", add=TRUE, what="model", lwd=c(1,1), addlegend = FALSE)
```

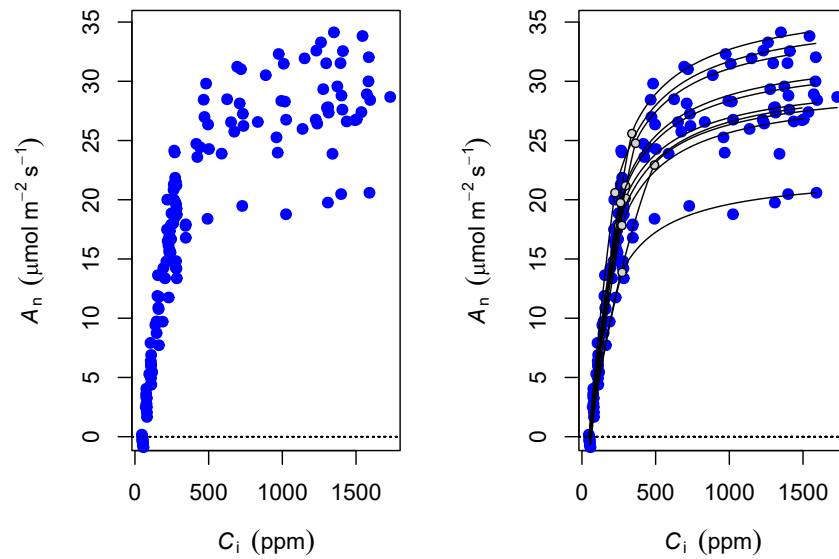


图 2.3: fitacis 作图结果

```
# 使用 sapply 提取拟合结果的 RMSE(均方根误差)
rmsees <- sapply(fits, "[", "RMSE")
plot(rmsees, type='h', ylab="RMSE", xlab="Curve nr")
# 对最差的拟合结果进行作图
plot(fits[[which.max(rmsees)])])
```

可以看出, *fitaci* 和 *fitacis* 用法基本一致, 各行代码均已经注释, 更详细用法请参考函数帮助。

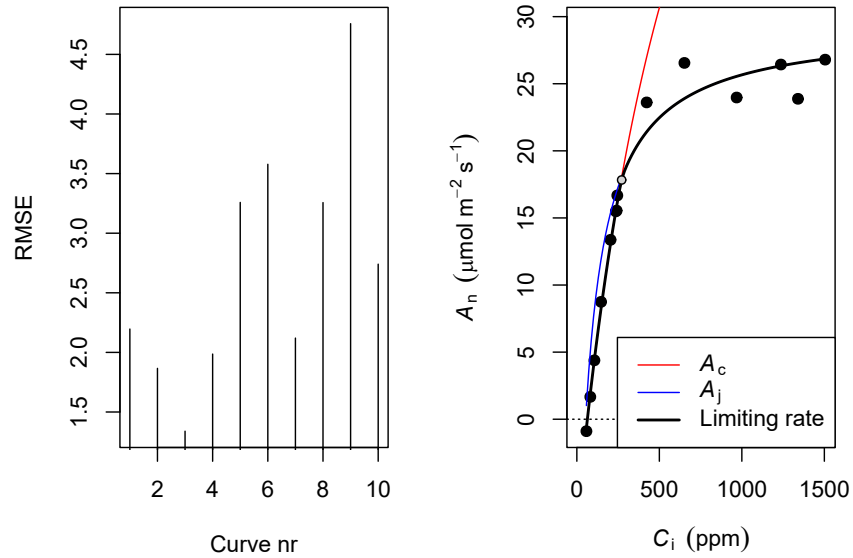


图 2.4: fitaci 作图结果

2.4.5 findCiTransition 函数

计算 CiTransition 的函数，第一点为 A_c & A_j ，第二点为 A_j & A_p ，并且仅在计算 TPU 的前提下才会有第二点出现。

```
findCiTransition(object, ...)
```

参数使用，object 为 fitaci 函数对象，或者整个的 Photosyn 函数。... 为使用 Photosyn 时可传递的参数。

2.5 C₄ 植物光合

之前的部分模型全部为关于 C₃ 植物的拟合，而 [Caemmerer \(2000\)](#) 的方法，则是针对 C₄ 植物的 A-Ci 曲线的实现。

```

AciC4(Ci, PPFD = 1500, Tleaf = 25, VPMAX25 = 120,
      JMAX25 = 400, Vcmax = 60, Vpr = 80,
      alpha = 0, gbs = 0.003, O2 = 210,
      x = 0.4, THETA = 0.7, Q10 = 2.3,
      RD0 = 1, RTEMP = 25, TBELOW = 0,
      DAYRESP = 1, Q10F = 2, FRM = 0.5, ...)

```

参数详解

- Ci: 胞间二氧化碳浓度 ($\mu\text{mol} \cdot \text{m}^{-2} \cdot \text{s}^{-1}$)。
- PPFD: 光合光子通量密度 ($\mu\text{mol} \cdot \text{m}^{-2} \cdot \text{s}^{-1}$)。
- Tleaf: 叶片温度 (°C)。
- VPMAX25: PEP 羧化最大速率 ($\mu\text{mol} \cdot \text{m}^{-2} \cdot \text{s}^{-1}$)。
- JMAX25: 最大电子传递速率 (°C))。
- Vcmax: 最大羧化速率 ($\mu\text{mol} \cdot \text{m}^{-2} \cdot \text{s}^{-1}$)。
- Vpr: PEP 再生 ($\mu\text{mol} \cdot \text{m}^{-2} \cdot \text{s}^{-1}$)。
- alpha: 维管束鞘细胞中 PSII 活性的比例。
- gbs: 维管束鞘导度 ($\text{mol} \cdot \text{m}^{-2} \cdot \text{s}^{-1}$)。
- O2: 叶肉细胞氧气浓度。
- x: 电子传递的分配因子。
- THETA: 曲角参数。
- Q10: Michaelis-Menten 系数中依赖于温度的参数。
- RD0: 基温下的呼吸 ($\text{mol} \cdot \text{m}^{-2} \cdot \text{s}^{-1}$)。
- RTEMP: 呼吸的基温 (°C)。
- TBELOW: 此温度以下呼吸为 0。
- DAYRESP: 明呼吸和暗呼吸的比值。

- Q10F: 呼吸依赖于温度的参数。
- FRM: 明呼吸中为叶肉呼吸的比例。

以上参数均来自 [Caemmerer \(2000\)](#), 括号中的参数值均为默认值, 具体应用时请按照实际情况修改。

2.5.1 C4 植物光合速率的计算

```
# 模拟 C4 植物的  $C_i$  值, 计算光合速率并作图  
library(plantecophys)  
aci <- AciC4(Ci=seq(5,600, length=101))  
with(aci, plot(Ci, ALEAF, type='l', ylim=c(0,max(ALEAF))))
```

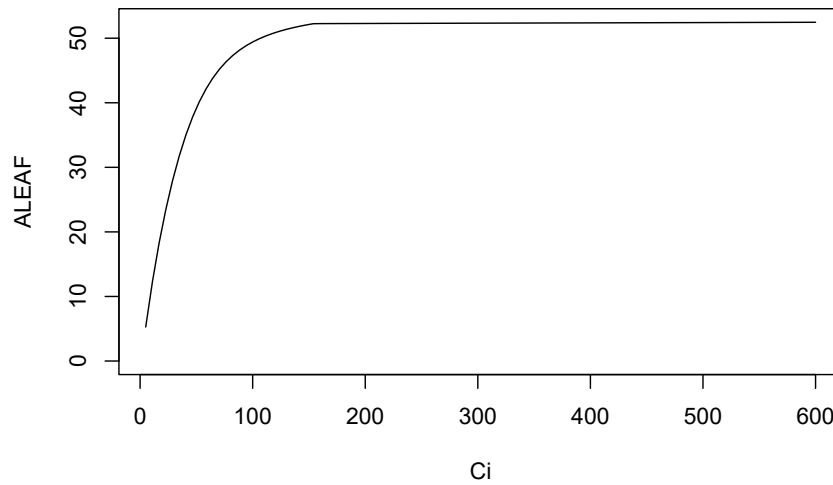


图 2.5: C4 植物 A- C_i 作图

3

气孔导度模型的拟合

气孔导度模型的拟合是通过 `fitBB` 来实现的，可以拟合三个 **Ball-Berry** 类型的气孔导度模型，共有下面几个参数：

- 气孔导度 (g_s),
- 光合 (A),
- 外界 CO_2 浓度 (C_a)
- 水气压亏缺 (VPD).

其三个模型的简介如下：

3.1 BallBerry 模型

[Ball et al. \(1987\)](#) 发表的文章中的模型：

$$g_s = g_0 + g_1 \left(\frac{Ah_r}{C_a} \right) \quad (3.1)$$

其中 A 为净光合速率， g_0 和 g_1 为拟合参数， h_r 为叶片表面的相对湿度， C_a 为叶片处 CO_2 浓度。

3.2 BBLeuning 模型

Leuning (1995) 发表的文章中的模型:

$$g_s = g_0 + g_1 \left(\frac{A}{(C_a - \Gamma)(1 + \frac{D}{D_0})} \right) \quad (3.2)$$

其中 Γ 为 CO_2 补偿点, g_0 、 g_1 和 D_0 为拟合参数。

3.3 BBOptiFull 模型

Medlyn et al. (2011) 发表的文章中的模型:

$$g_s^* \approx g_0 + g_1 \left(1 + \frac{g_1}{D} \right) \frac{A}{C_a} \quad (3.3)$$

额外的参数 g_k 来自于 Duursma et al. (2013)

$$g_s = g_0 + 1.6 \left(1 + \frac{g_1}{D} (1 - g_k) \right) \frac{A}{C_a} \quad (3.4)$$

3.4 fitBB 函数

```
fitBB(data, varnames = list(
  ALEAF = "A", GS = "gsw", VPD = "VPDleaf",
  Ca = "CO2_s", RH = "RHcham"),
  gsmodel = c("BBOpti", "BBLeuning", "BallBerry",
    "BBOptiFull"), fitg0 = FALSE)
```

参数的意义:

- data: 待分析的数据文件。
- varnames: 注意, 函数默认数据为 6400 格式, 因此 6800 的数据文件要安装上文的参数修改。相对湿度只有在使用 BallBerry 时才需要输入。
- gsmodel: 上述三个模型之一。
- fitg0: 默认不计算 g0, 若需要, 改为 TRUE。

代码示例:

```
library(plantecophys)

aci <- read.csv("../data/aci.csv")
aci <- subset(aci, Obs > 0)
fitBB(aci, gsmodel = "BBOpti", fitg0 = TRUE)

## Result of fitBB.
## Model : BBOpti
## Both g0 and g1 were estimated.
##
## Coefficients:
## g0 g1
## 0.326 -0.992
##
## For more details of the fit, look at summary(myfit$fit)
## To return coefficients, do coef(myfit).
## (where myfit is the name of the object returned by fitBB)
```

3.5 fitBBs 函数

如果我们有多个物种的数据，那么使用 `fitBBs` 则可以快速拟合多条曲线的数据。我们先整合两次的数据，然后再查看运行结果：

```
aci01 <- read.csv("../data/aci01.csv")
aci01 <- subset(aci01, Obs > 0)
multiBB <- data.frame(
  A = c(aci$Photo, aci01$Photo),
  GS = c(aci$Cond, aci01$Cond),
  CO2S = c(aci$CO2S, aci01$CO2S),
  VPD = c(aci$VpdL, aci01$VpdL),
  RH = c(aci$RH_S, aci01$RH_S),
  species = c(rep("species1", length(aci$Photo)),
    rep("species2", length(aci01$Photo)))
)

mod2 <- fitBBs(multiBB, group = "species",
  varnames = list(
    ALEAF = "A", GS = "GS", VPD = "VPD",
    Ca = "CO2S", RH = "RH"),
  gsmodel = "BallBerry", fitg0 = TRUE)
```

```
## RH provided in % converted to relative units.
## RH provided in % converted to relative units.
```

```
coef(mod2)
```

```
##      group      g0      g1
```

3.5 *fitBBs* 函数

31

```
## 1 species1 0.32638852 -0.1734554  
## 2 species2 0.05158725 -0.0218842
```



4

光合最优气孔导度耦合模型

Cowan and Farquhar (1977) 关于最优气孔导度的描述概括如下：最优气孔行为理论认为气孔的最优化行为就是在某一时间段内，最大化光合碳固定的同时最小化蒸腾作用，也就是说，对于一定的水分消耗，最大化光合碳固定。即使得：

$$A - \lambda E \quad (4.1)$$

有最大化，其中 λ 是临界水分利用效率，也即植物损耗单位水分的 C 生产量，单位一般为 $\text{mol CO}_2 \cdot \text{mol}^{-1} \text{H}_2\text{O}$ ，可通过光合速率和蒸腾速率计算。

4.1 FARAO 函数

FARAO 函数用于找到最大化 $A - \lambda E$ 的 ** Ci ** 值。

```
FARAO(lambda = 0.002, Ca = 400, VPD = 1,
      photo = c("BOTH", "VCMAX", "JMAX"),
      energybalance = FALSE, C4 = FALSE,
      Tair = 25, Wind = 2, Wleaf = 0.02,
      StomatalRatio = 1, LeafAbs = 0.86, ...)

FARAO2(lambda = 0.002, Ca = 400,
      energybalance = FALSE, ...)
```

其参数同 `fitaci` 和 `Photosyn`, 在此不多做介绍, 可参考 ([Cowan and Farquhar, 1977](#), [Buckley et al. \(2014\)](#), [Medlyn et al. \(2011\)](#))。

5

光合气孔导度耦合模型

Duursma (2015) 对于气体交换耦合模型的简述如下¹:

- 对于 FvCB 模型有:

$$A_n = \min(A_c, A_j) - R_d \quad (5.1)$$

- 在假定 g_m 为恒定的前提下:

$$C_c = C_i - \frac{A_n}{g_m} \quad (5.2)$$

- 根据 Ficker 定理

$$A_n = \frac{g_s}{1.6}(C_a - C_i) \quad (5.3)$$

以及前文 3.1 提到的气孔导度模型

$$g_s = g_0 + g_1 \frac{A_n}{C_a} f(D) \quad (5.4)$$

整合公式 (5.1), (5.2), (5.3) 和 (5.4), 气体交换耦合模型有很多更大尺度上的应用, 例如 Duursma and Medlyn (2012) 和 wang1998a, 可用于预测 A_n , g_s 和蒸腾速率对主要环境驱动因子的响应 (除土壤水分), 并包含了主要的叶片性状 (g_1 , V_{cmax} , J_{max} , R_d 以及他们的温度依赖性)。

¹详细内容请参考原文

5.1 Photosyn 函数

Photosyn 为耦合的光合-气孔导度模型，基于 Farquhar 光合模型和 Ball-Berry 气孔导度模型。

```
Photosyn(VPD = 1.5, Ca = 400, PPFD = 1500,
         Tleaf = 25, Patm = 100, RH = NULL,
         gsmodel = c("BBOpti", "BBLeuning",
                     "BallBerry", "BBdefine"),
         g1 = 4, g0 = 0, gk = 0.5, vpdmin = 0.5,
         D0 = 5, GS = NULL, BBmult = NULL,
         alpha = 0.24, theta = 0.85, Jmax = 100,
         Vcmax = 50, gmeso = NULL, TPU = 1000,
         alphag = 0, Rd0 = 0.92, Q10 = 1.92,
         Rd = NULL, TrefR = 25, Rdayfrac = 1,
         EaV = 82620.87, EdVC = 0, delsC = 645.1013,
         EaJ = 39676.89, EdVJ = 2e+05, delsJ = 641.3615,
         GammaStar = NULL, Km = NULL, Ci = NULL,
         Tcorrect = TRUE, returnParsOnly = FALSE,
         whichA = c("Ah", "Amin", "Ac", "Aj"))

Aci(Ci, ...)
```

因为是光合气孔导度模型的耦合，故而参数与之前的函数相同，参见 2.4.2 和 3.4 部分的内容。

5.1.1 Photosyn 使用举例

```
library(plantecophys)
# 仅使用下面几个参数运行模型
# (其他参数使用默认值)
# 利用已测量或计算的参数
Photosyn(VPD=2, g1=4, Ca=500)
```

```
##          Ci    ALEAF      GS    ELEAF      Ac      Aj    Ap    Rd VPD
## 1 369.3981 14.19466 0.1706377 3.412753 15.12654 17.03685 1000 0.92 2
## Tleaf Ca      Cc PPFD Patm
## 1    25 500 369.3981 1500 100
```

```
# 部分参数相同，而某一参数或某几个参数不同时，
# 可以将不同的参数设置为一个序列 (vectors)
r <- Photosyn(VPD=seq(0.5, 4, length=25),
              Vcmax=50, Jmax=100)
with(r, plot(VPD, ALEAF, type='l'))
```

不同 VPD 下的光合速率见 5.1。

```
# 设定叶肉导度的拟合
run1 <- Photosyn(PPFD=seq(50,1000,length=25),
                 gmeso=0.15, Vcmax=40, Jmax=85)
with(run1, plot(PPFD, GS, type='l'))
```

```
# 运行 ACi 曲线数据 (提供 Ci 值而不是计算)
arun1 <- Aci(Ci=seq(50, 1200, length=101),
             Vcmax=40, Jmax=85)
arun2 <- Aci(Ci=seq(50, 1200, length=101),
```

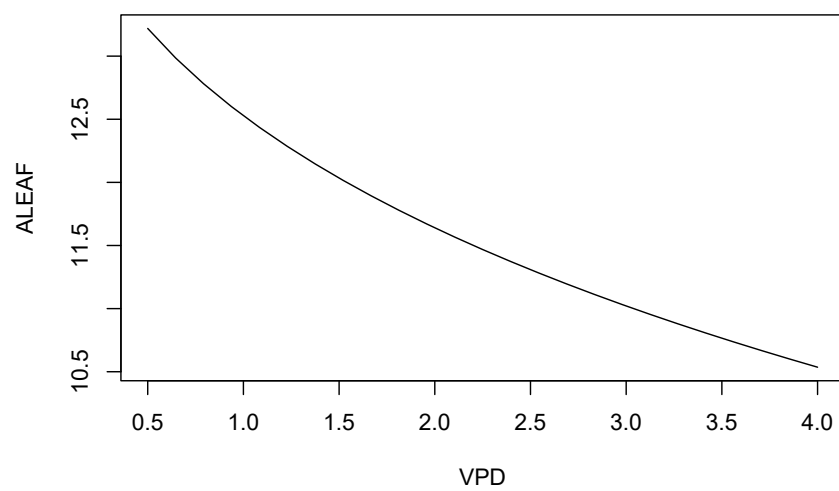


图 5.1: VPD VS. A_n

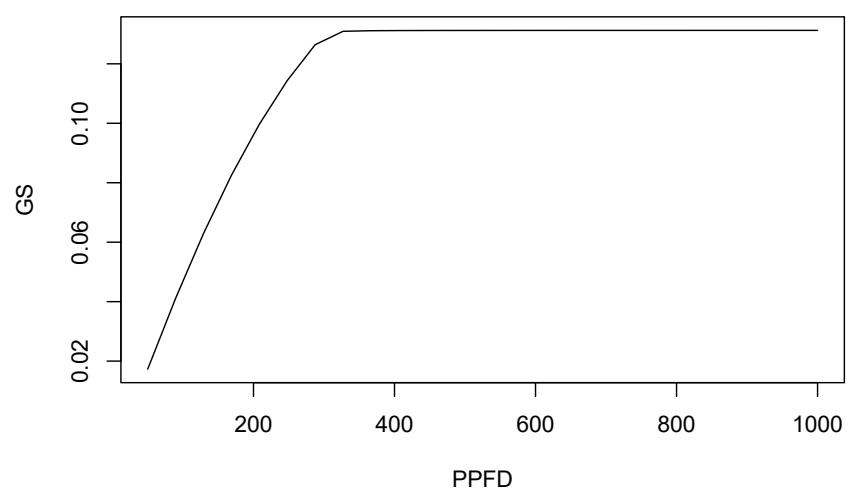


图 5.2: PPFD VS. GS

```
Vcmax=30, Jmax=70)
with(arun1, plot(Ci, ALEAF, type='l'))
with(arun2, points(Ci, ALEAF, type='l', lty=5))
```

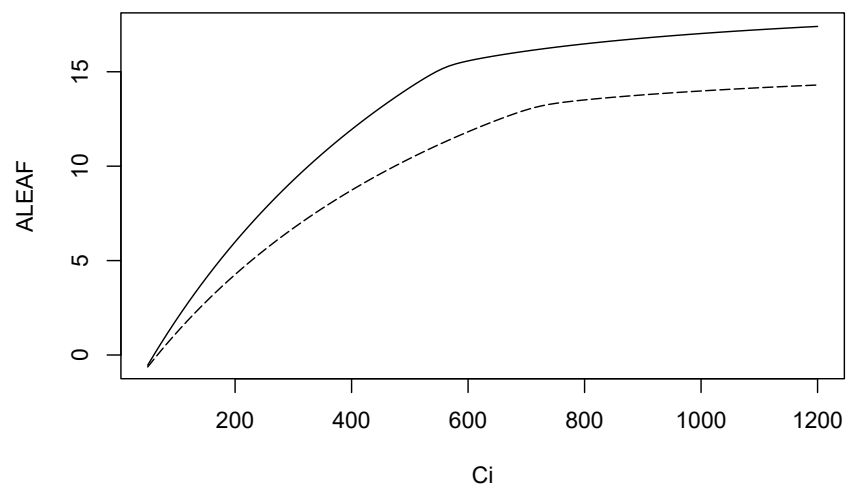


图 5.3: PPFD VS. GS

不同 ci 下的光合速率见 5.3。

```
# 找出 CO2 需求和供给的交叉点
# 设定部分参数
gs <- 0.2 # 对水的气孔导度
Ca <- 400 # 外部 CO2
gctogw <- 1.57 # 转换系数
gc <- gs / gctogw # 对 CO2 的气孔导度
# 需求曲线
p <- Aci(seq(60,500,length=101), Ca=400)
```

```

# 提供气孔导度及交叉点
g <- Photosyn(GS=gs, Ca=Ca)
# 交叉点可视化
par(yaxs="i")
with(p, plot(Ci, ALEAF, type='l',
             ylim=c(0,max(ALEAF))))
with(g, points(Ci, ALEAF, pch=19, col="red"))
abline(gc * Ca, -gc, lty=5)
legend("topleft", c(expression(
  "Demand: ~A==f(C[i]),
  expression("Supply: ~A==g[c]*(C[a]-C[i])),
  "Operating point"),
  lty=c(1,5,-1),pch=c(-1,-1,19),
  col=c("black","black","red"),
  bty='n', cex=0.9)

```

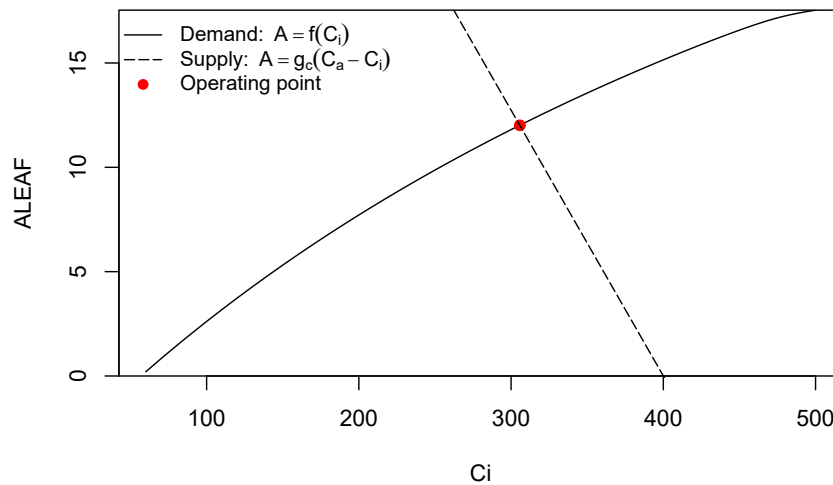


图 5.4: supply VS. demand

需求与供给的作图 5.4。

5.2 PhotosynEB 函数

使用同 Photosyn, 只不过使用能量平衡法来计算叶温。目前版本尚存在 bug, 不能直接提供 GS 来计算, 否则会出现未知错误。

5.3 PhotosynTuzet 函数

同样为光合气孔导度耦合模型, 只不过基于 Tuzet et al. (2003) 的气孔导度模型, 现将其描述如下:

$$g_{co2} = g_0 + \frac{aA}{C_i - \Gamma} f_{\Psi_v} \quad (5.5)$$

其中 g_{co2} 为对 CO_2 的气孔导度, g_0 是残余导度 (residual conductance) (在补偿点时限制 g_{CO_2} 的值), A 为光合速率, C_i 为胞间二氧化碳浓度, Γ 为 CO_2 补偿点, a 是经验系数。

一个根据经验的逻辑方程用于描述气孔对水势的敏感性:

$$f_{\Psi_v} = \frac{1 + \exp(s_f \Psi_f)}{1 + \exp(s_f (\Psi_f - \Psi_v))} \quad (5.6)$$

Ψ_v 是指叶片水势, Ψ_f 是参比势能, 该方程假设在水势接近于 0 时, 气孔对叶片水势不敏感, 并且气孔随着 Ψ_v 的下降快速关闭。 Ψ_f 和 s_f (曲线形状参数) 依赖于不同物种的形态学适应以及生长过程中所处的环境。

5.3.1 PhotosynTuzet 的参数

除 Tuzet et al. (2003) 模型中使用的参数外, 其他参数都继承了 Photosyn 的参数

```
PhotosynTuzet(g1 = 8, Ca = 400, psis = 0,  
              kl = 2, sf = 3, psif = -2,  
              ...)
```

- g1: 斜率参数, 要远比 fitBB 中的大。
- Ca: 外界大气 CO₂ 浓度
- psis, 土壤水势 (Mpa)。
- kl: Leaf-specific hydraulic conductance (叶片导水参数 ($\text{mmol} \cdot \text{m}^{-2} \cdot \text{s}^{-1} \cdot \text{MPa}^{-1}$), 叶片蒸腾量、叶片面积及叶片水势计算)。
- sf: 曲线形状参数。
- 在气孔导度为最大值 50% 时的叶片水势。

6

*RH*to*VPD* 函数

该函数为一系列的工具，用于温度、湿度不同单位之间的换算 [Jones \(1993\)](#)：

```
# RH 转为 VPD
RHtoVPD(RH, TdegC, Pa = 101)

# VPD 转为 RH
VPDtoRH(VPD, TdegC, Pa = 101)

# 饱和水气压计算
esat(TdegC, Pa = 101)

# 露点温度转换为 VPD
DewtoVPD(Tdew, TdegC, Pa = 101)

# 基于叶温的 VPD 转换为基于空气温度的 VPD
VPDleafToAir(VPD, Tleaf, Tair, Pa = 101)

# 基于空气温度的 VPD 转换为基于叶温的 VPD
VPDairToLeaf(VPD, Tair, Tleaf, Pa = 101)

# 基于叶温的相对湿度转换为基于空气温度的相对湿度
RHleafToAir(RH, Tleaf, Tair, Pa = 101)
```

```
# 基于空气温度的相对湿度转换为基于叶温的相对湿度  
RHairToLeaf(RH, Tair, Tleaf, Pa = 101)
```

注意事项及单位：

部分参数的转换需要准确的大气压用于计算，因此，有大气压选项的参数必须填入实际值。

RH: 相对湿度 (%)。

TdegC: 温度 (°C) (叶片或温度)。

Pa: 大气压 (kPa)

VPD: 水气压亏缺 (kPa)。

Tdew: 露点温度 (°C)。

Tleaf: 叶温 (°C)。

Tair: 空气温度 (°C)。

7

光响应曲线的拟合

光响应曲线模型有很多，主要分为四大类，直角双曲线，非直角双曲线，指数以及直角双曲线修正模型，我们分别对这四类进行阐述。

7.1 直角双曲线模型

Baly (1935) 提出了直角双曲线模型，它的表达式为：

$$P_n = \frac{\alpha I P_{nmax}}{\alpha I + P_{nmax}} - R_d \quad (7.1)$$

- 其中， P_n 为净光合速率；
- I 为光强；
- α 为光响应曲线在光强为 0 时的斜率，即光响应曲线的初始斜率，也称为初始量子效率；
- P_{nmax} 为最大净光合速率；
- R_d ：为暗呼吸速率。

对 (7.1) 求导可知其导数大于 0，也就是直角双曲线是一个没有极值的渐近线，因此，无法由 (7.1) 求得最大光合速率的饱和光强¹。

因此就需要使用弱光条件下 ($\leq 200 \mu mol \cdot m^{-2} \cdot s^{-1}$) 的数据得到表观量子

¹直角双曲线和非直角双曲线模型类似，如果测量时饱和光强之后光合速率不下降，则计算的 P_{nmax} 则远大于实测值。

效率 (apparent quantum efficiency, AQE), 利用非线性最小二乘法估算 P_{nmax} , 然后利用 ZiPiao (2010) 的式 (7.2) 求解 I_{sat} ,

$$P_{nmax} = AQE \times I_{sat} - R_d \quad (7.2)$$

但此方法测得的光饱和点远小于实测值, 我们采用 $0.7P_{nmax}$ Zhang et al. (2009)、 $0.9P_{nmax}$ Huang et al. (2009)、或其他设定的值来的来估算 I_{sat} 。

7.1.1 直角双曲线模型的实现

```
# 调用非线性拟合包 minpack.lm, 也可以直接使用 nls
library(minpack.lm)
# 读取数据, 同 fitaci 数据格式
lrc <- read.csv("../data/lrc.csv")
lrc <- subset(lrc, Obs > 0)

# 光响应曲线没有太多参数,
# 直接调出相应的光强和光合速率
# 方便后面调用
lrc_Q <- lrc$PARi
lrc_A <- lrc$Photo

# 采用非线性拟合进行数据的拟合
lrcnls <- nlsLM(lrc_A ~ (alpha * lrc_Q * Am) *
               (1/(alpha * lrc_Q + Am)) - Rd,
               start=list(Am=(max(lrc_A)-min(lrc_A)),
                           alpha=0.05,Rd=-min(lrc_A))
               )
fitlrc_rec <- summary(lrcnls)

# 补偿点时净光合速率为 0,
```

```
# 据此利用 uniroot 求解方程的根
Ic <- function(Ic){(fitlrc_rec$coef[2,1] * Ic *
  fitlrc_rec$coef[1,1]) * (1/(fitlrc_rec$coef[2,1] *
  Ic + fitlrc_rec$coef[1,1])) - fitlrc_rec$coef[3,1]
}

uniroot(Ic, c(0,50))$root

## [1] 3.650053
```

```
# 根据饱和点定义, 0.75 最大光合速率为饱和点,
# 也可以是其他比例
# 据此利用 uniroot 求解方程的根

Isat <- function(Isat){(fitlrc_rec$coef[2,1] *
  Isat * fitlrc_rec$coef[1,1]) *
  (1/(fitlrc_rec$coef[2,1] * Isat +
    fitlrc_rec$coef[1,1])) - fitlrc_rec$coef[3,1] -
  0.75 * fitlrc_rec$coef[1,1]
}

# 求值区间根据具体实验确定

uniroot(Isat, c(0,2500))$root

## [1] 700.0946
```

```
# 使用 ggplot2 进行作图并拟合曲线
library(ggplot2)
```

```

light <- data.frame(lrc_Q = lrc$PARi, lrc_A = lrc$Photo)

p <- ggplot(light, aes(x = lrc_Q, y = lrc_A))

p1 <- p + geom_point(shape = 16, size = 3, color = "green") +
  geom_smooth(method="nls", formula = y ~ (alpha * x * Am) *
    (1/(alpha * x + Am)) - Rd, se = FALSE,
    method.args =
      list(start = c(Am=(max(lrc_A)-min(lrc_A)),
        alpha=0.05,Rd=-min(lrc_A)),
        aes(x =lrc_Q, y = lrc_A, color='blue', size = 1.2))
  ) +
  labs(y=expression(paste("photosynthetic rate ",
    "(", mu, mol%.%m^-2%.%s^-1, ")")),
    x=expression(paste("PAR ",
    "(", mu, mol%.%m^-2%.%s^-1, ")")))

# 自定义坐标轴
p1 + scale_x_continuous(breaks = seq(0, 2100, by = 200)) +
  scale_y_continuous(breaks= round(light$lrc_A)) +
  theme(axis.text.x = element_text(
    size = 10, angle=30, vjust=0.5),
    axis.text.y = element_text(size = 10),
    axis.title.x = element_text(size = 12, face = 'bold'),
    axis.title.y = element_text(size = 12, face = 'bold')
  )

```

代码目的见注释，其实现过程主要分三步：

- 数据的导入，这与之前相同，具体格式方法参考前文 ??。

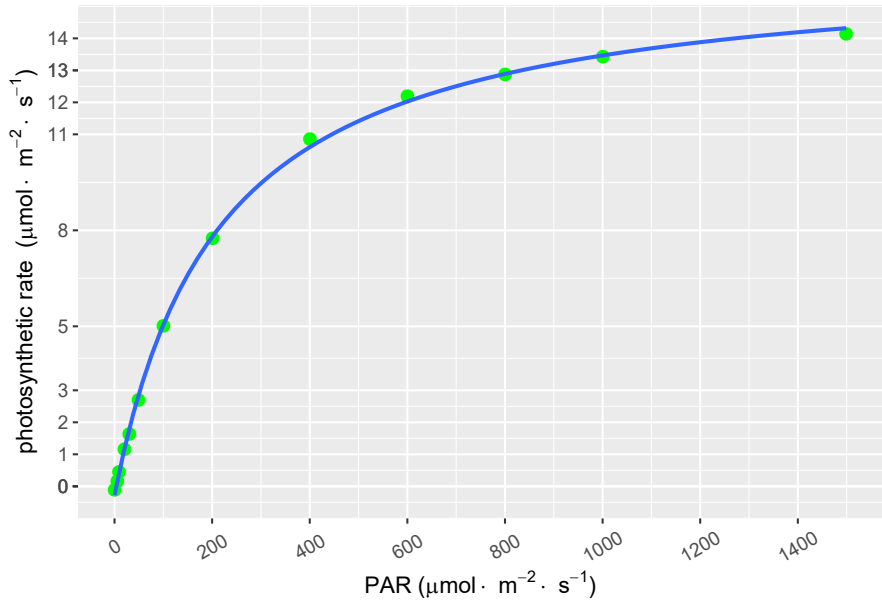


图 7.1: 直角双曲线模型拟合

表 7.1: 直角双曲线计算参数

	Estimate	Std. Error	t value	Pr(> t)
Am	16.6721752	0.1522849	109.480151	0.0000000
alpha	0.0783312	0.0026774	29.256870	0.0000000
Rd	0.2810926	0.0789338	3.561117	0.0051716

- 光响应曲线的拟合，使用到了非线性模型 nlsLM，也可以使用 nls，具体实现方法请查看参考文档。
- 求饱和点和补偿点，补偿点的计算根据其定义，净光合速率为 0，求解模型在一定区间的根来计算，而饱和点则较为麻烦，若使用式 (7.2) 计算，那么饱和点远远低于我们实际需求的，因此，我们使用了 $0.75P_{nmax}$ 来计算，求得目标区间的根。当然也可以采用其他比例来作为饱和点光合速率。

最终的数据拟结果如图 7.1 所示，拟合的参数及结果见表 7.1。



7.2 非直角双曲线模型

Thornley (1976) 提出了非直角双曲线模型，它的表达式为：

$$P_n = \frac{\alpha I + P_{nmax} \sqrt{(\alpha I + P_{nmax})^2 - 4\theta \alpha I P_{nmax}}}{2\theta} - R_d \quad (7.3)$$

其中， θ 为表示曲线弯曲程度的曲角参数，取值为 $0 \leq \theta \leq 1$ 。其他参数意义同式 (7.1)。同样如同直角双曲线模型，式仍然没有极值，无法求得 I_{sat} ，可以仍然参考直角双曲线模型的方式进行计算。

7.2.1 非直角双曲线模型的实现

```
library(minpack.lm)

# 读取数据，同 fitaci 数据格式
lrc <- read.csv("../data/lrc.csv")
lrc <- subset(lrc, Obs > 0)

# 光响应曲线没有太多参数，
# 直接调出相应的光强和光合速率
# 方便后面调用
lrc_Q <- lrc$PARi
lrc_A <- lrc$Photo

# 非直角双曲线模型的拟合
lrcnls <- nlsLM(lrc_A ~
                (1/(2*theta))*
                (alpha*lrc_Q+Am-sqrt((alpha*lrc_Q+Am)^2 -
                4*alpha*theta*Am*lrc_Q))- Rd,
```

```

start=list(Am=(max(lrc_A)-min(lrc_A)),
alpha=0.05,Rd=-min(lrc_A),theta=1))

fitlrc_nrec <- summary(lrcnls)

# 光补偿点
Ic <- function(Ic){
  (1/(2 * fitlrc_nrec$coef[4,1])) *
  (fitlrc_nrec$coef[2,1] * Ic + fitlrc_nrec$coef[1,1] -
  sqrt((fitlrc_nrec$coef[2,1] * Ic + fitlrc_nrec$coef[1,1]
  )^2 - 4 * fitlrc_nrec$coef[2,1] *
  fitlrc_nrec$coef[4,1] * fitlrc_nrec$coef[1,1] * Ic)) -
  fitlrc_nrec$coef[3,1]
}

uniroot(Ic, c(0,50))$root

```

```
## [1] 2.234292
```

```

# 光饱和点
Isat <- function(Isat){
  (1/(2 * fitlrc_nrec$coef[4,1])) * (fitlrc_nrec$coef[2,1] *
  Isat + fitlrc_nrec$coef[1,1] - sqrt(
  (fitlrc_nrec$coef[2,1] * Isat + fitlrc_nrec$coef[1,1])^2 -
  4*fitlrc_nrec$coef[2,1] * fitlrc_nrec$coef[4,1] *
  fitlrc_nrec$coef[1,1] * Isat)) -
  fitlrc_nrec$coef[3,1] - (0.9*fitlrc_nrec$coef[1,1])}

uniroot(Isat, c(0,2000))$root

```

```
## [1] 1596.286
```

```

# 使用 ggplot2 进行作图并拟合曲线
library(ggplot2)
light <- data.frame(lrc_Q = lrc$PARi, lrc_A = lrc$Photo)

p <- ggplot(light, aes(x = lrc_Q, y = lrc_A))

p1 <- p + geom_point(shape = 16, size = 3, color = "green") +
  geom_smooth(method="nls", formula = y ~
    (1/(2*theta))*(alpha*x+Am-sqrt((alpha*x+Am)^2 -
    4*alpha*theta*Am*x))- Rd, se = FALSE,
    method.args = list(start = c(Am=(max(lrc_A)-min(lrc_A)),
    alpha=0.05, Rd=-min(lrc_A), theta=1),
    aes(x =lrc_Q, y = lrc_A, color='blue', size = 1.2))
  ) +
  labs(y=expression(paste("photosynthetic rate ",
    "(", mu, mol%.%m^-2%.%s^-1, ")")),
    x=expression(paste("PAR ",
    "(", mu, mol%.%m^-2%.%s^-1, ")")))

# 自定义坐标轴
p1 + scale_x_continuous(breaks = seq(0, 2100, by = 200)) +
  scale_y_continuous(breaks= round(light$lrc_A)) +
  theme(axis.text.x = element_text(
    size = 10, angle=30, vjust=0.5),
    axis.text.y = element_text(size = 10),
    axis.title.x = element_text(size = 12, face = 'bold'),
    axis.title.y = element_text(size = 12, face = 'bold')
  )

```

最终的数据拟结果如图 7.2 所示，拟合的参数及结果见表 7.2。单纯从作图来看，本例数据使用非直角双曲线与散点图重合程度更高。

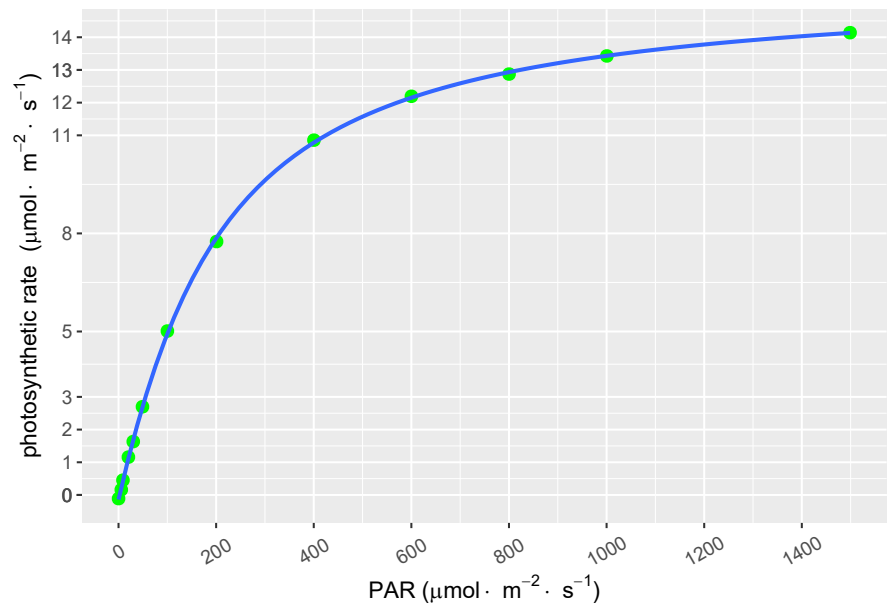


图 7.2: 非直角双曲线模型拟合

表 7.2: 非直角双曲线计算参数

	Estimate	Std. Error	t value	Pr(> t)
Am	15.8017296	0.1513064	104.435285	0.0000000
alpha	0.0658067	0.0020216	32.551422	0.0000000
Rd	0.1461717	0.0420800	3.473659	0.0070082
theta	0.3700908	0.0493403	7.500783	0.0000369

7.3 指数模型

光合指数模型较多，我们此处使用的指数函数的模型 [Prado and Moraes \(1997\)](#)，其表达式为：

$$P_n = P_{nmax}[1 - e^{-b(I-I_c)}] \quad (7.4)$$

其中， I_c 为光补偿点， e 为自然对数的底， b 为常数，其他参数意义同 (7.4)。同样，该方程仍然是没有极值的函数，但我们可以直接求得光补偿点。

7.3.1 指数模型的实现

```
library(minpack.lm)

# 读取数据，同 fitaci 数据格式
lrc <- read.csv("../data/lrc.csv")
lrc <- subset(lrc, Obs > 0)

# 光响应曲线没有太多参数，
# 直接调出相应的光强和光合速率
# 方便后面调用
lrc_Q <- lrc$PARi
lrc_A <- lrc$Photo

# 模型的拟合
lrcnls <- nlsLM(lrc_A ~ Am*(1-exp((-b)*(lrc_Q-Ic))),
               start=list(Am=(max(lrc_A)-min(lrc_A)),
                           Ic=5, b=1)
               )
fitlrc_exp <- summary(lrcnls)
```

```
# 光饱和点
Isat <- function(Isat){fitlrc_exp$coef[1,1]*
  (1-exp((-fitlrc_exp$coef[3,1])*(Isat-
    fitlrc_exp$coef[2,1]))) - 0.9*fitlrc_exp$coef[1,1]}

uniroot(Isat, c(0,2000))$root
```

```
## [1] 558.6038
```

```
## 拟合图形
library(ggplot2)
light <- data.frame(lrc_Q = lrc$PARi, lrc_A = lrc$Photo)

p <- ggplot(light, aes(x = lrc_Q, y = lrc_A))

p1 <- p +
  geom_point(shape = 16, size = 3, color = "green") +
  geom_smooth(method="nls", formula =
    y ~ Am*(1-exp((-b)*(x -Ic))),
    se = FALSE, method.args = list(
      start = c(Am=(max(lrc_A)-min(lrc_A)),
        Ic=5, b=0.002), aes(x =lrc_Q, y = lrc_A,
          color='blue', size = 1.2))
    ) +
  labs(y=expression(paste("photosynthetic rate ",
    "(", mu, mol%.%m^-2%.%s^-1, ")")),
    x=expression(paste("PAR ",
    "(", mu, mol%.%m^-2%.%s^-1, ")")))
```



```
# 自定义坐标轴
p1 + scale_x_continuous(breaks = seq(0, 2100, by = 200)) +
  scale_y_continuous(breaks = round(light$lrc_A)) +
  theme(axis.text.x = element_text(
    size = 10, angle=30, vjust=0.5),
    axis.text.y = element_text(size = 10),
    axis.title.x = element_text(size = 12, face = 'bold'),
    axis.title.y = element_text(size = 12, face = 'bold')
  )
)
```

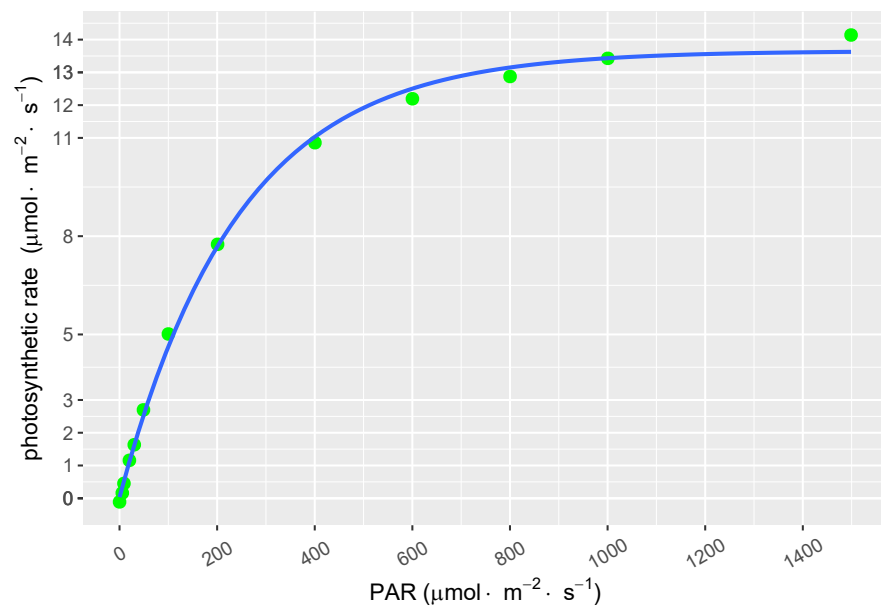


图 7.3: 指数模型拟合

最终的数据拟结果如图 7.3 所示，拟合的参数及结果见表 7.3。

表 7.3: 指数模型计算参数

	Estimate	Std. Error	t value	Pr(> t)
Am	13.6547568	0.1723363	79.233185	0.0000000
Ic	-0.5133438	2.3370250	-0.219657	0.8305573
b	0.0041183	0.0002012	20.467032	0.0000000

7.4 直角双曲线的修正模型

ZiPiao (2010) 直角双曲线修正模型的表达式如式 (7.5) 所示:

$$P_n = \alpha \frac{1 - \beta I}{1 + \gamma I} I - R_d \quad (7.5)$$

其中, β 和 γ 为系数, β 光抑制项, γ 光饱和项, 单位为 $m^2 \cdot s \cdot \mu mol^{-1}$, 其他参数与上文相同, 因为该式 (7.5) 存在极值, 因此, 必然存在饱和光强和最大净光合速率, 分别用式 (7.6) 和式 (7.7) 求得。

$$I_{sat} = \frac{\sqrt{\frac{(\beta + \gamma)}{\beta}} - 1}{\gamma} \quad (7.6)$$

$$P_{nmax} = \alpha \left(\frac{\sqrt{\beta + \gamma} - \sqrt{\beta}}{\gamma} \right)^2 - R_d \quad (7.7)$$

该模型的优点为拟合结果中光饱和点和最大净光合速率均接近实测值, 还可以拟合饱和光强之后光合速率随光强下降段的曲线。

7.4.1 直角双曲线修正模型的实现

```
library(minpack.lm)

# 读取数据, 同 fitaci 数据格式
lrc <- read.csv("../data/lrc.csv")
lrc <- subset(lrc, Obs > 0)

# 光响应曲线没有太多参数,
# 直接调出相应的光强和光合速率
# 方便后面调用
lrc_Q <- lrc$PARi
```

```

lrc_A <- lrc$Photo

# 模型的拟合
lrcnls <- nlsLM(lrc_A ~ alpha * ((1 -
    beta*lrc_Q)/(1 + gamma * lrc_Q)) * lrc_Q - Rd,
    start=list(alpha = 0.07, beta = 0.00005,
    gamma=0.004, Rd = 0.2)
)
fitlrc_mrec <- summary(lrcnls)

# 饱和点计算
Isat <- sqrt((fitlrc_mrec$coef[2,1] + fitlrc_mrec$coef[3,1])/
    fitlrc_mrec$coef[2,1]-1)/fitlrc_mrec$coef[3,1]

# 补偿点计算
Ic <- (fitlrc_mrec$coef[1,1] - fitlrc_mrec$coef[4,1])/
    (fitlrc_mrec$coef[3,1]*fitlrc_mrec$coef[4,1]+
    fitlrc_mrec$coef[1,1]*fitlrc_mrec$coef[2,1])

## 拟合图形
library(ggplot2)
light <- data.frame(lrc_Q = lrc$PARi, lrc_A = lrc$Photo)

p <- ggplot(light, aes(x = lrc_Q, y = lrc_A))

p1 <- p +
    geom_point(shape = 16, size = 3, color = "green") +
    geom_smooth(method="nls", formula =
    y ~ alpha * ((1 -
        beta*x)/(1 + gamma * x)) * x - Rd,
    se = FALSE, method.args = list(
    start = c(alpha = 0.07, beta = 0.00005,
    gamma=0.004, Rd = 0.2),

```

表 7.4: 直角双曲线修正模型计算参数

	Estimate	Std. Error	t value	Pr(> t)
alpha	0.0730858	0.0021209	34.460183	0.0000000
beta	0.0000501	0.0000133	3.776115	0.0043751
gamma	0.0040622	0.0001955	20.773916	0.0000000
Rd	0.2156186	0.0543505	3.967190	0.0032685

```

aes(x =lrc_Q, y = lrc_A,
    color='blue', size = 1.2))
) +
labs(y=expression(paste("photosynthetic rate ",
    "(", mu, mol%.%m^-2%.%s^-1, ")")),
    x=expression(paste("PAR ",
    "(", mu, mol%.%m^-2%.%s^-1, ")")))

# 自定义坐标轴
p1 + scale_x_continuous(breaks = seq(0, 2100, by = 200)) +
    scale_y_continuous(breaks= round(light$lrc_A)) +
    theme(axis.text.x = element_text(
        size = 10, angle=30, vjust=0.5),
        axis.text.y = element_text(size = 10),
        axis.title.x = element_text(size = 12, face = 'bold'),
        axis.title.y = element_text(size = 12, face = 'bold')
    )

```

尽管修正模型可以方便的计算饱和点和补偿点,但如同 Lobo et al. (2013) 所指出,双曲线模型对其结果的计算常有超出生态学意义范围的值²,因此对模型的选择不能一概而论,需根据实际情况而选择。

²例如本例的数据结果

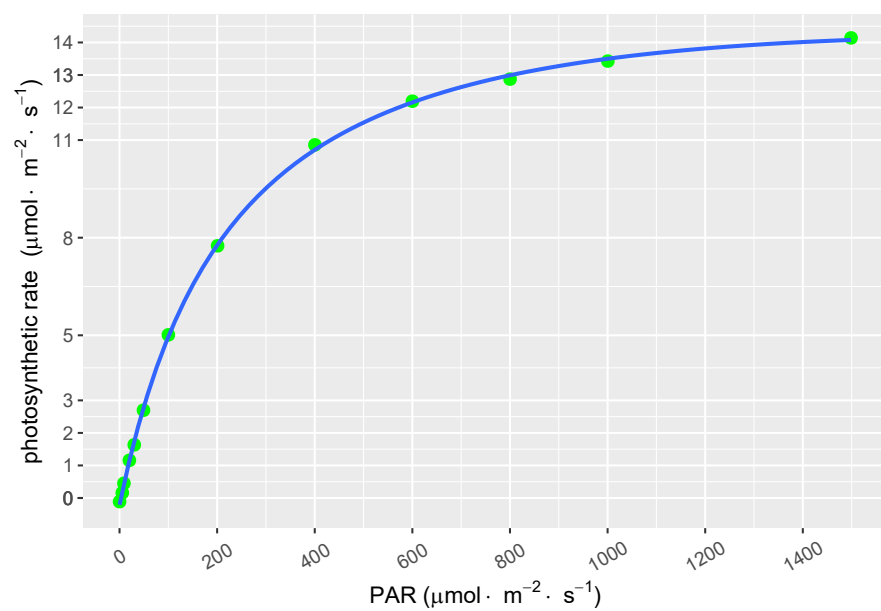


图 7.4: 直角双曲线修正模型拟合

8

关于非线性拟合的初始值

在解释初始值之前我们首先需要了解一个数学上的概念——迭代，

“迭代法”也称“辗转法”是一种不断用变量的旧值递推新值的过程。

用通俗但不是特别严谨的说法可解释为：每次执行这种算法是，程序都会从原值（也就是我抄的上面迭代法定义的旧值）推出一个新值。

之所以先介绍这个迭代，原因很简单，非线性拟合就是通过迭代的方法，需要对每一个变量最初的估计值进行不断的迭代，得到一个向一个点收缩或汇聚的值，这个估计值必须在实际值的一定范围内，程序通过不断调整这个值来改善拟合结果。这就解释了上面的问题，初始值是让程序开始运行的前提，不然没法迭代，必须设定。我下面的内容将以 LI-6800 的光响应曲线的测试数据，使用非直角双曲线模型进行拟合来讲解具体的 R 中的一些实现方法，我们首先导入数据，然后再利用这些数据逐个举例不同的确定初始值的方式。

```
nls <- read.csv("data/nlstest.csv")
```

```
# 光响应曲线比较简单，我们将需要的数据直接提取，方便后面操作
```

```
lrc_Q <- nls$Qin
lrc_A <- nls$A
```

8.1 nlsLM 解决方案

nlsLM 来自于 [Elzhov et al. \(2016\)](#) 的 `minpack.lm`, 利用 C 语言的 MINPACK 库, 修改了 Levenberg-Marquardt 算法, 在实际操作中, 很多时候并不准确的输入初始值, 他也能得出比较好的拟合结果。但结果未必完美, 出现下面让人烦恼的报错:

```
singular gradient matrix at initial parameter estimates
```

的概率会大大降低, 而且尽管结果不如意, 我们也可以利用他的结果缩小初始值的范围, 继续尝试其他初始值。

例如下面的例子中, 非直角双曲线的 `Rd` 的初始值我们可以利用暗呼吸的实测值大致估计, 同理最大光合速率也是如此, 剩下的分别为非直角双曲线速率, 我们暂定为 1, `alpha` 也暂定为 0.1, 使用 `nlsLM` 进行拟合, 结果如下:

```
library(minpack.lm)

lrcnls_lm <- nlsLM(lrc_A ~ (1/(2*theta))*
  (alpha*lrc_Q+Am-sqrt((alpha*lrc_Q+Am)^2 -
  4*alpha*theta*Am*lrc_Q))-
  Rd, start=list(Am=(max(lrc_A)-min(lrc_A)),
  alpha=0.1,Rd=-min(lrc_A),theta=0.8))
```


结果没有报错，看上去没有问题，那我们观察一下具体的拟合结果：

```
summary(lrcnls_lm)
```

```
##
## Formula: lrc_A ~ (1/(2 * theta)) * (alpha * lrc_Q + Am - sqrt((alpha *
##      lrc_Q + Am)^2 - 4 * alpha * theta * Am * lrc_Q)) - Rd
##
## Parameters:
##      Estimate Std. Error t value Pr(>|t|)
## Am      12.307570   0.406739  30.259 2.30e-10 ***
## alpha   0.045706   0.003423  13.352 3.09e-07 ***
## Rd       0.656638   0.132646   4.950 0.000791 ***
## theta   0.707522   0.079738   8.873 9.59e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1852 on 9 degrees of freedom
##
## Number of iterations to convergence: 8
## Achieved convergence tolerance: 1.49e-08
```

结果看上去还可以¹。

8.2 作图比对法

模型很多参数可以用已有数据去估计，我们可以只来分析难以判断的参数，流程如下：

¹ 有些时候结果并不理想，该方法并不是万无一失

- Rd、Am 等我们可以利用测量值来确定一个范围。
- 剩余的参数，我们也可以根据经验或文献来有一个大致的判断。
- 然后我们根据数学的方式来判断哪个参数对曲线形状影响最大（例如在分母上的参数，或者是乘以该参数，该参数可以显著改变计算结果，例如整体乘以或除以 0.1 还是 0.01，像 Rd 之类的参数本身就很小，多数公式中都是减去该值，对结果影响很小，我们通常直接使用实测值）。
- 将该参数取一系列值带入模型来求解净光合速率。
- 将计算的 A 值与光强进行作图，看我们计算的曲线与测量数据点的重合程度，必要时在修改其他参数，使曲线和散点重合度最好，重合程度最高的参数值即为我们需要的初始值。

8.2.1 实现过程

```
# 我们选择的模型，将其写为一个函数，用于计算净光合速率
expfct <- function(x, Am, alpha, Rd, theta) {
  (1/(2 * theta)) * (alpha * x + Am -
    sqrt((alpha * x + Am)^2 - 4 * alpha * theta * Am * x)) - Rd
}

# 我们的数据
test <- data.frame(x = lrc_Q, y = lrc_A)
```

```
# 先做实测数据的散点图
plot(y ~ x, data = test)

# 利用上面的函数，假定 alpha 的值为 0.8，看计算值与测量值重合程度
curve(expfct(x, Am = (max(lrc_A)-min(lrc_A)),
  alpha=0.8, Rd=-min(lrc_A), theta=0.8), add = TRUE
)
```

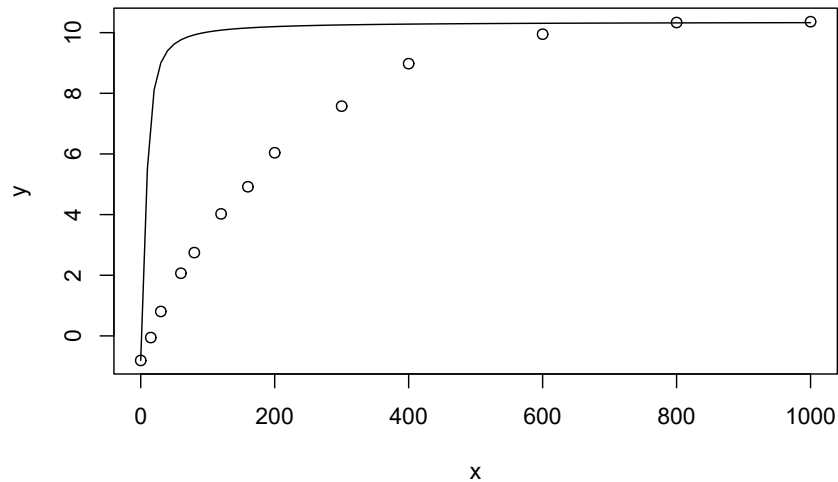
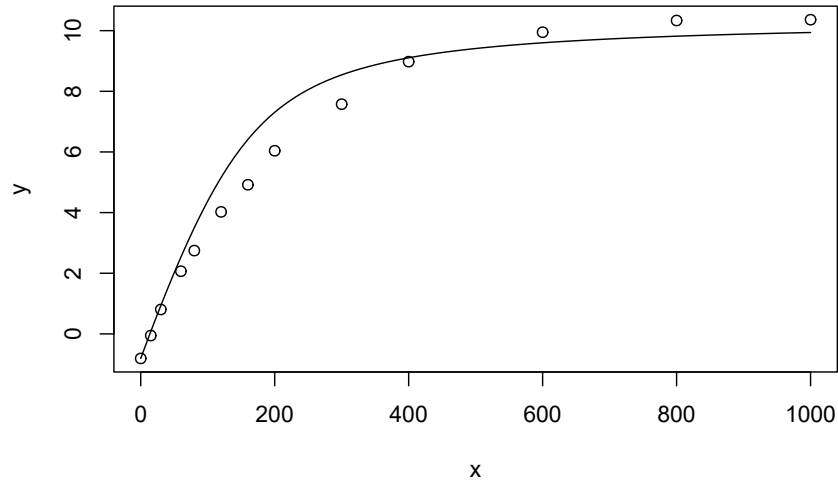


图 8.1: 初步判断 alpha 的初始值

观察上图 8.1 的结果可以看到，曲线在 0-600 的范围内，拟合值明显偏大，观察模型的方程式，以及其他起始值的设定方式，我们初步判断 alpha 的值偏大，于是乎我们将其改小观察，观察曲线和测量点的重合仍然不是很好，我们尝试修改 theta 值与 alpha 值（也即曲线高于测量点，则需要减小纵坐标的值，低于测量点，则需要增加该值，该过程省略，我大概设置了五分钟完成），最终得出的结果如下：

```
plot(y ~ x, data = test)
curve(expfct(x, Am = (max(lrc_A)-min(lrc_A)),
              alpha=0.06, Rd=-min(lrc_A), theta=0.82), add = TRUE)
```

图 8.2 尽管看上去效果仍然不满意，但我们可试着进行拟合，看能否得到显著差异的结果：

图 8.2: 初修正后断 α 的初始值

```
lrcnls_manual <- nls(lrc_A ~
  (1/(2*theta))*
  (alpha*lrc_Q+Am-sqrt((alpha*lrc_Q+Am)^2 -
    4*alpha*theta*Am*lrc_Q))-
  Rd, start=list(Am=(max(lrc_A)-min(lrc_A)),
    alpha=0.03,Rd=-min(lrc_A),theta=0.6))
summary(lrcnls_manual)
```

```
##
## Formula: lrc_A ~ (1/(2 * theta)) * (alpha * lrc_Q + Am - sqrt((alpha *
##   lrc_Q + Am)^2 - 4 * alpha * theta * Am * lrc_Q)) - Rd
##
## Parameters:
##      Estimate Std. Error t value Pr(>|t|)
## Am      12.307585   0.406741  30.259 2.30e-10 ***
```

```
## alpha  0.045706    0.003423   13.352 3.09e-07 ***
## Rd      0.656642    0.132646    4.950 0.000791 ***
## theta   0.707518    0.079739    8.873 9.59e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1852 on 9 degrees of freedom
##
## Number of iterations to convergence: 7
## Achieved convergence tolerance: 4.601e-06
```

```
# 对拟合之后的结果作图，观察使用我们的估计值，
# 迭代的最终值与元数据的重合程度
plot(y ~ x, data = test, ylim = c(-2, 14))
curve(expfct(x, Am = 12.307586,
  alpha=0.045706, Rd= 0.656643, theta=0.707518), add = TRUE)
```

从 8.3 的呈现以及 F 检验的 p 值来讲，图形已经比较完美了。也就是说尽管我们作图的时候看到重合度并不高，但是非线性拟合本来就是一个迭代的过程，只要我们的数据与真实值相差不大，还是能够得到完美结果的。

8.2.2 直观展示

上面的表述太啰嗦，直接用下面的图形说明一下，其中 α 的取值在此处选择从 0.01 到 0.07，每次增加 0.05，其他值分别为 $Am = 12.31$, $Rd = 0.66$, $\theta = 0.71$ （此处为展示效果和方便，将这些值直接按照拟合结果设定了，实际差别不大）

```
library(ggplot2)
library(purrr)
```

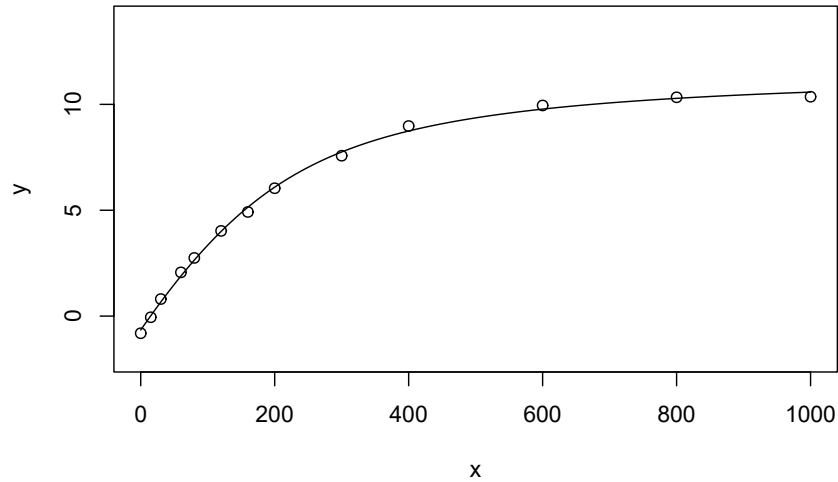


图 8.3: 检验作图法的初始值判断

```
lrc <- read.csv("data/nlstest.csv")

# 光响应曲线比较简单，我们将需要的数据直接提取，方便后面操作
lrc_Q <- lrc$Qin
lrc_A <- lrc$A
n <- length(lrc_A)

alp <- paste0("a=", seq(0.01, 0.07, by = 0.005))
alpn <- rep(alp, each = n)

expfct <- function(x, Am, alpha, Rd, theta) {(1/(2 * theta)) * (alpha * x + Am - sqrt(
})

paras <- data.frame(alpha = rep(seq(0.01, 0.07, by = 0.005), each = n),
```

```

      x = rep(lrc_Q, n), Am = rep(12.31, n), Rd = rep(0.66, n),
      theta = rep(0.71, n))
y = unlist(pmap(paras, expfct))

show <- data.frame(x = rep(lrc_Q, 14),
  y = c(lrc_A, y),
  a = factor(c(rep("measured", n), alpn),
  level = c("measured", alp)
  ))

ggplot(data = show, aes(x, y, group = a, color=a)) +
  geom_point() +
  geom_smooth(se = FALSE)

```

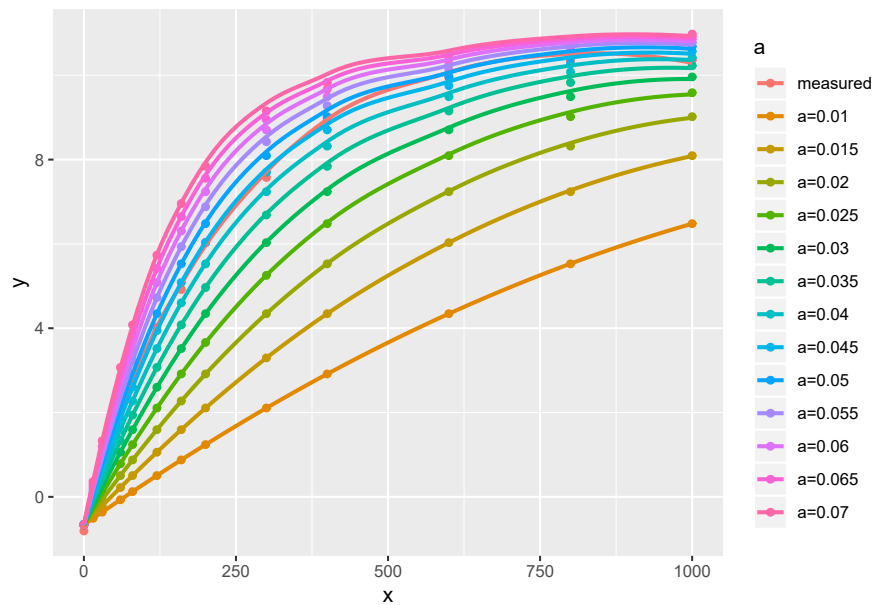


图 8.4: 多个 alpha 取值的差异

从上图 8.4 我们我们可以看到，实测值在 $\alpha = 0.04$ 和 $\alpha = 0.05$ 两条

曲线之间，在 0.045 时最接近测量点，也就是我们把初始值设为 0.04 和 0.05 之间最接近，本例中可认为是 0.045，实际这三个值均可。

8.3 自动多次尝试法

该方法实际为使用 `nls2` 来实现，具体方法参考 [Bouvier and Huet \(1994\)](#) 的文章，可简单概括为使用一系列的起始值梯度（例如下面的代码中，`alpha` 的取值在 0.01 到 0.08 之间），然后软件循序使用不同的起始值，即排列组合所有的起始值序列，最终找到合适的值，具体实现如下：

```
library(nls2)

grid.test <- expand.grid(list(
  Am=c(12),
  alpha = seq(0.01, 0.08, by =0.01),
  Rd = seq(0, 3),
  theta=seq(0.1, 1, by = 0.1)
))

lrcnls2 <- nls2(lrc_A ~
  (1/(2*theta))*
  (alpha*lrc_Q+Am-sqrt((alpha*lrc_Q+Am)^2 -
    4*alpha*theta*Am*lrc_Q))-
  Rd, start = grid.test, algorithm = "brute-force")
summary(lrcnls2)

##
## Formula: lrc_A ~ (1/(2 * theta)) * (alpha * lrc_Q + Am - sqrt((alpha *
##      lrc_Q + Am)^2 - 4 * alpha * theta * Am * lrc_Q)) - Rd
##
```



```
## Parameters:
##      Estimate Std. Error t value Pr(>|t|)
## Am      12.000000    0.623023  19.261 1.27e-08 ***
## alpha    0.050000    0.006414   7.795 2.72e-05 ***
## Rd       1.000000    0.260153   3.844 0.00394 **
## theta    0.800000    0.102143   7.832 2.62e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3663 on 9 degrees of freedom
##
## Number of iterations to convergence: 320
## Achieved convergence tolerance: NA
```

通过结果可以看到，虽然和之前采用手动方法判定的结果比较接近，但是还是略有差异，可以看一下他们各自的结果同测量值的重合程度：

```
plot(lrc_Q, lrc_A)
lines(lrc_Q, predict(lrcnls2), col="red")
lines(lrc_Q, predict(lrcnls_manual), col="blue")
```

图 8.5 可以看到，使用 `nls2` 的拟合结果似乎和测量值更匹配，当然这只是第一印象，后续的判断还要进一步通过 F 检验、AIC、BIC 等统计方式才能判定。

8.4 小结

采用如上三种方式都可以有效的解决起始值的问题，`nlsLM` 操作上更易实现，对初始值的大小不敏感，但设置不能太离谱，否则仍然会报错。作图比对法操作上更麻烦一些，但是这种方式一定能得出合理的初始值设置。采用 `nls2`

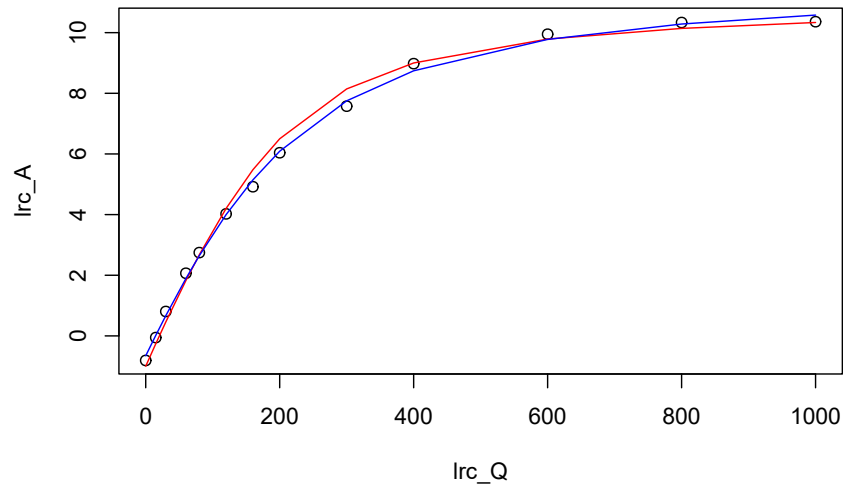


图 8.5: 两种方法结果的对比展示

类似于将手动作图方式自动化，类似于 SPSS 中非线性拟合中需要给出一个初始值的范围，且该范围不能过大。如有一定的经验，操作起来将非常迅速。

需要注意的是，这三种方法结合起来使用会更好，例如，即使使用 `nlsLM` 的结果不合理，也可以参考他们参数的范围（部分结果也可能是差异显著），然后将这些结果用于手动作图判定参数或者 `nls2` 中判定参数范围，或者使用作图法确定大致的范围，将该范围输入到 `nls2` 中，这样会节省时间，也更加方便。

9

LI-6800 的数据分析

9.1 数据格式

LI-6800 的数据前处理同 LI-6400 相同，如表 9.1 所示的样式。

表 9.1: 推荐 LI-6800 整理后数据样式

obs	time	elapsed	date	TIME	E
1	1471425750	0.0	20160817 17:22:30	1471425726	0.0007614
2	1471425855	104.5	20160817 17:24:14	1471425829	0.0009221
3	1471426153	403.1	20160817 17:29:13	1471426144	0.0009900
4	1471426244	494.0	20160817 17:30:44	1471426235	0.0010249
5	1471426335	585.1	20160817 17:32:15	1471426326	0.0010632
6	1471426426	676.0	20160817 17:33:46	1471426417	0.0011190

9.2 LI-6800 与 LI-6400 使用时的差别

plantecophys 使用时建立在 LI-6400XT 基础之上的软件包，因此在 LI-6800 代码中，需要改动的是 fitaci、fitacis 及 fitBB 中的 varnames 选项，也就是将 LI-6400XT 的表头改为 LI-6800 的表头。以 fitaci 函数为例：

```
fitaci(aci, varnames =  
  list(ALEAF = "A", Tleaf = "Tleaf", Ci = "Ci",  
        PPFD = "Qin", Rd = "Rd"))
```

9.3 光响应曲线注意事项

光响应曲线的拟合相对简单，仅需要光强和光合速率的值，其中需要修改的部分仅为光强的赋值部分，在文件名一致的前提下，修改如下代码即可：

```
lrc_Q <- lrc$Qin  
lrc_A <- lrc$A
```

9.4 LI-6800 RACiR™ 的测量与拟合

在评估作物性状时， V_{cmax} 及 J_{max} 时非常有用，传统的 A-Ci 曲线测量要求植物叶片要在一定浓度 CO_2 下适应几分钟后完成测量，这样的测量有几个缺点：

- 测量时间长，一条曲线至少需要 20 – 30 min，样本量多，重复多时，这种方法几乎没有可行性。
- 整个测量过程中，时间长，酶的激活状态会有变化，叶绿体会移动，气孔的开度也会发生变化。

而 LI-6800 独有的 auto control 功能在算法上允许用户自定义 CO_2 的起始浓度和种植浓度、变化方式（线性或其他）、所花费的时间，再加上其 IRGAs 极快的响应频率，使得短时间内的 A-Ci 的测量成为现实，即快速 CO_2 响应

曲线 RACiR™ 测量实验，该功能使得 5 min 内测量 A-Ci 曲线成为可能。该方法的实现可参考 Stinziano et al. (2017) 的文章。

Stinziano et al. (2018) 针对 RACiR™ 技术的疑问做了解答并提出了准确测量的建议，概括如下：

- 首先，采用 100 ppm/min 的变化速率是与标准方法重合度最高的测量。
- 其次，明确研究问题，目前已有研究表明 V_{cmax} 与 J_{max} 的计算结果与标准测量方法结果无显著差异。
- 任何条件的改变，都需要做空叶室校准，例如：流速，气体浓度变化方向、温度，斜率等。
- 空叶室校准与叶片测量采用严格的同一次校准，因为 IRGA 的漂移，需要再次匹配时，或者环境条件改变时，需要重新做空叶室校准。是否需要匹配，可通过不加叶片的最初状态查看，此时 A 值应接近为 0，reference 和 sample 气体浓度读数接近相等。
- IRGA 分析器使用 5 此多项式进行校准，推荐使用 1 次到 5 次多项式进行拟合，然后根据 BIC 指数来确定最合适的空叶室校准系数（即非参数拟合的模型选择的问题）。确定最合适的浓度变化范围。通常需要去掉最初和最后 30 s 的数据。
- 最小化校准和测量值之间的水分摩尔分数差异。甚至有可能需要控制 reference 或 sample 的水的摩尔分数而不是 V_{pdleaf} 。通过预实验来确定合适的 CO_2 变化范围和随时间的斜率。

同时 Stinziano et al. (2018) 还给出了专门的 RACiR 数据分析软件包 racir package¹，以下内容是我之前写的内容，部分代码的实现和上述软件包相似，仅供参考。

后续会更新 racir 相关的分析方法。

¹<https://github.com/jstinzi/racir>

9.5 RACiR™ 分析的实现

```
# 分别读取未校准空叶室数据
# 未校准带叶片测量数据
# 标准 aci 曲线测量数据
uemp500 <- read.csv("./data/uncorr_emp500.csv")
uleaf500 <- read.csv("./data/uncorr_leaf500.csv")
acin <- read.csv("./data/aci_ex.csv")

# 防止读入空白行
m <- length(which(uemp500$obs >= 1))
n <- length(which(uleaf500$obs >= 1))
uemp500 <- uemp500[1:m,]
uleaf500 <- uleaf500[1:n,]

# 观察空叶室未校准数据 reference 对 A 的图形
plot(uemp500$C02_r, uemp500$A)
# 选取线性部分用于校准
locator()
# 执行 locator 命令后, 在上图中的目标位置选点,
# 选好后按 esc 可以返回所选点的坐标 (选点即为在
# 预期位置鼠标单击)

# 根据上面的点, 利用二氧化碳的值过滤掉不需要的数据
# 只要在线性范围内选点, 拟合结果相差很小
cemp <- uemp500[which(uemp500$C02_r >
                      45.28 & uemp500$C02_r < 459.12),]
plot(cemp$C02_r, cemp$A)

cemp1st <- lm(A ~ C02_r, data = cemp)
```

```

comp2nd <- lm(A ~ poly(CO2_r, 2), data = comp)
comp3rd <- lm(A ~ poly(CO2_r, 3), data = comp)
comp4th <- lm(A ~ poly(CO2_r, 4), data = comp)
comp5th <- lm(A ~ poly(CO2_r, 5), data = comp)

bics <- BIC(cal1st, cal2nd, cal3rd, cal4th, cal5th)
best <- noquote(rownames(bics)[bics$BIC == min(bics$BIC)])

best

# 校准带叶片测量的数据
uleafc <- uleaf500
uleafc$A <- uleafc$A - predict(cal4th, uleafc)
uleafc$Ci <- ((uleafc$gtc - uleafc$E / 2) * uleafc$CO2_s -
              uleafc$A) / (uleafc$gtc + uleafc$E / 2)

# 对校准前后的数据进行作图，查看校准效果
plot(uleaf500$CO2_r, uleaf500$A, pch = 2, ylim = c(-20, 40))
points(uleafc$CO2_r, uleafc$A)
locator()
cleaf <- uleafc[which(uleafc$CO2_r > 16.6 &
                     uleafc$CO2_r < 478),]
plot(cleaf$CO2_r, cleaf$A)

# 利用 plantecophys 拟合标准曲线和 racir 曲线
library("plantecophys")
acifit <- fitaci(cleaf, varnames =
                list(ALEAF = "A", Tleaf = "Tleaf", Ci = "Ci",
                     PPFD = "Qin", Rd = "Rd"), Patm = 84.09)
acifit_normal <- fitaci(acin, varnames =
                      list(ALEAF = "A", Tleaf = "Tleaf", Ci = "Ci",
                           PPFD = "Qin", Rd = "Rd"), Patm = 84.09)

```

```

# 查看拟合数据
acifit$pars
acifit_normal$pars

# 对快速曲线作图拟合结果进行查看
plot(acifit, linecols = c("green", "blue", "red"))

#### ggplot2 作图
ddata <- acifit$df

alldata <- data.frame(A = c(acin$A, ddata$Ac, ddata$Aj, cleaf$A),
                      Ci = c(acin$Ci, ddata$Ci, ddata$Ci, cleaf$Ci),
                      Atype = c(
                        rep("standard Aci curve", length(acin$A)),
                        rep("Ac", length(ddata$Ac)),
                        rep("Aj", length(ddata$Aj)),
                        rep("A corrected", length(cleaf$A))
                      )
)

alldata$Atype <- factor(alldata$Atype,
                       levels = c("standard Aci curve",
                                   "Ac", "Aj", "A corrected"))

library(ggplot2)
p <- ggplot(alldata)
p1 <- p + geom_point(aes(Ci, A, colour = Atype), alpha = 0.5)

p2 <- p1 + geom_smooth(aes(Ci, A, colour = Atype),
                      method = 'auto') +
  labs(y=expression(paste("A ", "(", mu,
                          mol%.%m^-2%.%s^-1, ")")),

```



```

x=expression(paste(C[i], " ",
                    "(" , mu, mol%.%mol-1, ")"))))

p2 + scale_x_continuous(limits=c(0, 1800),
                        breaks=seq(0, 1800, 200)) +
scale_y_continuous(limits=c(-3, 50),
                  breaks=seq(0, 50, 10)) +
scale_colour_hue(name=" ",
                 labels=c('ACi curve data; ',
                           expression(paste(A[c], ' data; ')),
                           expression(paste(A[j], ' data; ')),
                           'RACiR data')
) + theme_set(theme_bw()) +
theme(axis.text.x = element_text(size = 10,
                                  angle=30, vjust=0.5),
      axis.text.y = element_text(size = 10),
      axis.title.x = element_text(
        size = 12, face = 'bold'),
      axis.title.y = element_text(
        size = 12, face = 'bold'),
      legend.text.align = 0,
      legend.position = c(0.8, 0.6))

```

最终采用常规方法和 RACiRTM 方法结果如图 9.1 所示，其中 A_j 与 A_c 结果采用 RACiRTM 计算²。

²注意，注意数据表头的大小写，此处代码中，为处理数据的方便，我更改了大小写，分析自己的数据时需要注意

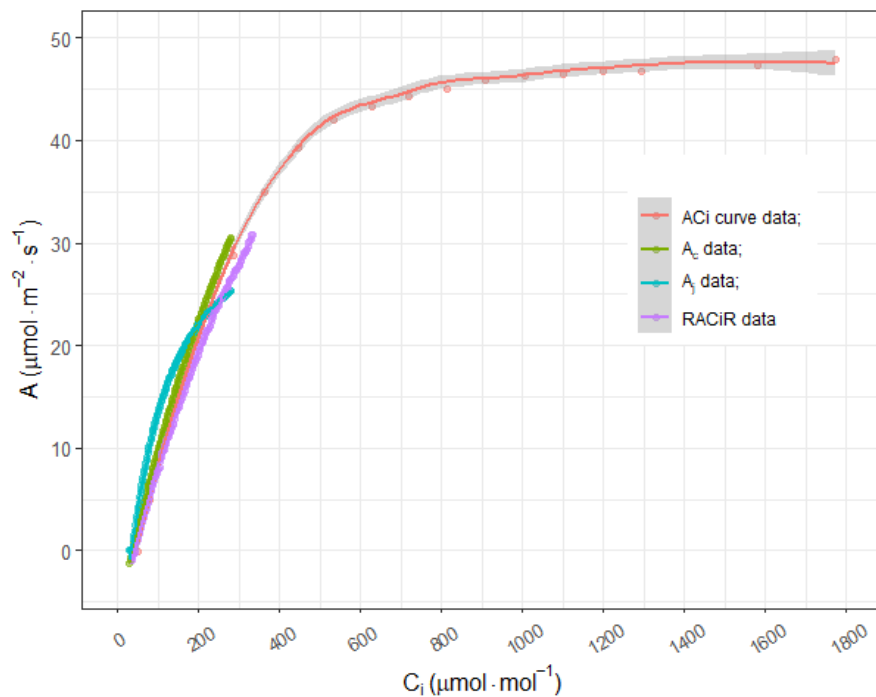


图 9.1: RACiR 方法与常规结果的比较

9.6 LI-6800 荧光数据分析

LI-6800 能够直接提供基本的叶绿素荧光参数，其他参数均可通过这些基本的参数进行计算，计算也较简单，在此不赘述，需要注意的是快相荧光部分的数据，因为分析 o_{jip} 数据的模型有很多，很多都需要复杂的计算，在此我们先将其中较为简单的 jip test 数据分析进行介绍。

9.6.1 jip test 的实现

LI-6800 增加了 o_{jip} 曲线测量功能，本功能主要是针对测量数据的 jip test 的实现。

表 9.2: jiptest 批量导入数据后的样式

SECS	FLUOR	SOURCE
2.41e-05	887	INDUCTION-484-20171225-13_15_58
2.81e-05	998	INDUCTION-484-20171225-13_15_58
3.19e-05	1013	INDUCTION-484-20171225-13_15_58
3.60e-05	1055	INDUCTION-484-20171225-13_15_58
4.01e-05	1044	INDUCTION-484-20171225-13_15_58
4.41e-05	1205	INDUCTION-484-20171225-13_15_58

9.6.2 jiptest 软件包安装

目前 `jiptest` 暂时放在 github 我的软件仓库内，并没提交 CRAN，因此需要 `devtools` 的支持，然后才能从 github 安装 `jiptest`。

```
install.packages("devtools")
library(devtools)
install_github("zhujiedong/jiptest")
```

9.6.3 read_files 及 read_dcfiles 函数

`read_files` 用于批量读取所有调制光测量数据，方便用于其他的数据分析。函数要求所有数据必须是 `xlsx` 格式，并且所有测量数据都保存在同一文件夹内。，如有其他文件或测量数据则会报错。`read_dcfiles` 用于批量读取所有连续光测量数据，其他与 `read_dcfiles` 相同。

函数仅有一个参数，即保存数据文件夹的路径，使用如下：

```
library(jiptest)
jipdata <- read_files("./data/ojip")
```

表 9.3: jiptest DC 数据批量导入数据后的样式

SECS	FLUOR	SOURCE
2.41e-05	154000	INDUCTION-484-20171225-13_15_58
2.81e-05	157566	INDUCTION-484-20171225-13_15_58
3.19e-05	161261	INDUCTION-484-20171225-13_15_58
3.60e-05	165114	INDUCTION-484-20171225-13_15_58
4.01e-05	169069	INDUCTION-484-20171225-13_15_58
4.41e-05	173231	INDUCTION-484-20171225-13_15_58

```
dcjipdata <- read_dcfiles("./data/ojip")
```

9.6.4 jip_test 及 jip_dctest 函数

jip_test 是 jiptest 的核心函数，用于所有数据的 jip test 分析，函数仅包含两个参数，测量数据的保存文件夹路径，以及分析结果的导出路径，导出路径必须为已存在的文件夹或磁盘根目录。jip_dctest 与 jip_test 相似，用于连续光测量数据的分析。

```
jip_results <- jip_test("./data/ojip", "d:/")
```

```
## for the current version, we suggest you use the default settings of the duration (
## in the induction settings of LI-6800
```

表 9.4: jiptest 输出的计算参数

parameters	data_file1	data_file2	data_file3	data_file4
Fo	887.0000000	849.0000000	981.0000000	914.0000000
Fm	4649.5200000	3714.2700000	4225.2400000	3754.2700000
F300	2181.4000000	1798.6000000	2067.8000000	2537.2000000

FJ	2594.3000000	2104.6000000	2456.8000000	2569.0000000
FI	4443.1100000	3547.1100000	4009.2200000	3556.5300000
Tfmax	256.0160000	316.0160000	272.0160000	200.0160000
Area	79.6780569	59.0630025	75.5884747	80.6896655
Fv	3762.5200000	2865.2700000	3244.2400000	2840.2700000
Vj	0.4537650	0.4382135	0.4548985	0.5826911
Mo	1.3760990	1.3256691	1.3399748	2.2859799
Sm	0.0211768	0.0206134	0.0232993	0.0284092
Ss	0.3297474	0.3305602	0.3394829	0.2548977
N	0.0642212	0.0623590	0.0686317	0.1114531
Vav	0.9999173	0.9999348	0.9999143	0.9998580
phi_po	0.8092276	0.7714221	0.7678238	0.7565439
phi_ET2o	0.4420284	0.4333745	0.4185419	0.3157125
phi_RE1o	0.0443938	0.0450048	0.0511261	0.0526707
psi_ET2o	0.5462350	0.5617865	0.5451015	0.4173089
psi_RE1o	0.0548595	0.0583401	0.0665857	0.0696201
delta_RE1o	0.1004321	0.1038474	0.1221529	0.1668312
phi_Do	0.1907724	0.2285779	0.2321762	0.2434561
phi_Pav	0.0000669	0.0000503	0.0000658	0.0001075
ABS_RC	3.7475545	3.9215460	3.8363703	5.1856106
TR0_RC	3.0326246	3.0251673	2.9456566	3.9231420
ET2o_RC	1.6565256	1.6994981	1.6056817	1.6371621
RE1o_RC	0.1663683	0.1764884	0.1961386	0.2731297
DI0_RC	0.7149299	0.8963787	0.8907137	1.2624686
ABC_CS	887.0000000	849.0000000	981.0000000	914.0000000
TR0_CS	3.0326246	3.0251673	2.9456566	3.9231420
ET0_CS	1.6565256	1.6994981	1.6056817	1.6371621
DI0_CS	883.9673754	845.9748327	978.0543434	910.0768580
PIabs	1.3625603	1.1032809	1.0329666	0.4291734
PItotal	0.1521228	0.1278497	0.1437378	0.0859364
DFabs	0.3093655	0.0982884	0.0324348	-0.8458942

```
dcjip_results <- jip_dctest("./data/ojip", "d:/")
```

```
## for the current version, we suggest you use the default settings of the duration (
##           in the induction settings of LI-6800
```

表 9.5: jiptest DC 数据输出的计算参数

parameters	data_file1	data_file2	data_file3	data_file4
Fo	1.540000e+05	1.331910e+05	1.566640e+05	1.535890e+05
Fm	7.138390e+05	5.606100e+05	6.476600e+05	5.848390e+05
F300	3.380910e+05	2.754768e+05	3.228592e+05	3.881104e+05
FJ	4.007440e+05	3.202450e+05	3.768850e+05	4.066710e+05
FI	6.815050e+05	5.343140e+05	6.145900e+05	5.522530e+05
Tfmax	2.760160e+02	3.000160e+02	2.880160e+02	2.120160e+02
Area	1.037380e+04	8.095231e+03	9.855470e+03	1.052059e+04
Fv	5.598390e+05	4.274190e+05	4.909960e+05	4.312500e+05
Vj	4.407410e-01	4.376361e-01	4.485189e-01	5.868568e-01
Mo	1.315314e+00	1.331581e+00	1.353943e+00	2.175271e+00
Sm	1.853000e-02	1.893980e-02	2.007240e-02	2.439560e-02
Ss	3.350843e-01	3.286589e-01	3.312686e-01	2.697856e-01
N	5.529940e-02	5.762750e-02	6.059250e-02	9.042570e-02
Vav	9.999329e-01	9.999369e-01	9.999303e-01	9.998849e-01
phi_po	7.842651e-01	7.624177e-01	7.581076e-01	7.373824e-01
phi_ET2o	4.386073e-01	4.287562e-01	4.180820e-01	3.046445e-01
phi_RE1o	4.529590e-02	4.690600e-02	5.106070e-02	5.571790e-02
psi_ET2o	5.592590e-01	5.623639e-01	5.514811e-01	4.131432e-01
psi_RE1o	5.775590e-02	6.152280e-02	6.735290e-02	7.556170e-02
delta_RE1o	1.032722e-01	1.094003e-01	1.221309e-01	1.828948e-01
phi_Do	2.157349e-01	2.375823e-01	2.418924e-01	2.626176e-01
phi_Pav	5.270000e-05	4.810000e-05	5.280000e-05	8.480000e-05
ABS_RC	3.805249e+00	3.990815e+00	3.981886e+00	5.026763e+00

TR0_RC	2.984324e+00	3.042668e+00	3.018699e+00	3.706647e+00
ET2o_RC	1.669010e+00	1.711086e+00	1.664755e+00	1.531376e+00
RE1o_RC	1.723623e-01	1.871934e-01	2.033181e-01	2.800807e-01
DIO_RC	8.209251e-01	9.481469e-01	9.631878e-01	1.320116e+00
ABC_CS	1.540000e+05	1.331910e+05	1.566640e+05	1.535890e+05
TR0_CS	2.984324e+00	3.042668e+00	3.018699e+00	3.706647e+00
ET0_CS	1.669010e+00	1.711086e+00	1.664755e+00	1.531376e+00
DIO_CS	1.539970e+05	1.331880e+05	1.566610e+05	1.535853e+05
PIabs	1.212241e+00	1.033289e+00	9.677646e-01	3.932321e-01
PItotal	1.396084e-01	1.269280e-01	1.346374e-01	8.801820e-02
DFabs	1.924705e-01	3.274650e-02	-3.276640e-02	-9.333551e-01

计算参数见表 9.4

9.6.5 jip_plot 及 jip_dcplot 函数

jip_plot 和 jip_dcplot 基于 ggplot2, 用于快速预览所有测量结果的 ojip 曲线。函数仅有一个参数，即保存数据文件夹的路径，使用如下：

```
jip_plot("./data/ojip")
```

```
jip_dcplot("./data/ojip")
```

ojip 调制式测量光 ojip 曲线的快速预览如图 9.2

ojip 连续式测量光 ojip 曲线的快速预览如图 9.3

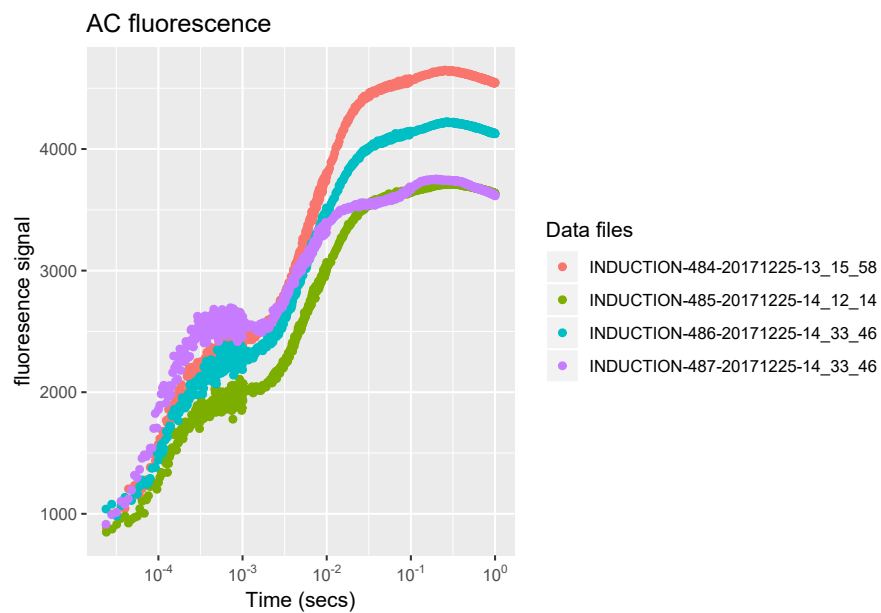


图 9.2: 调制式测量的 ojip 曲线的快速预览

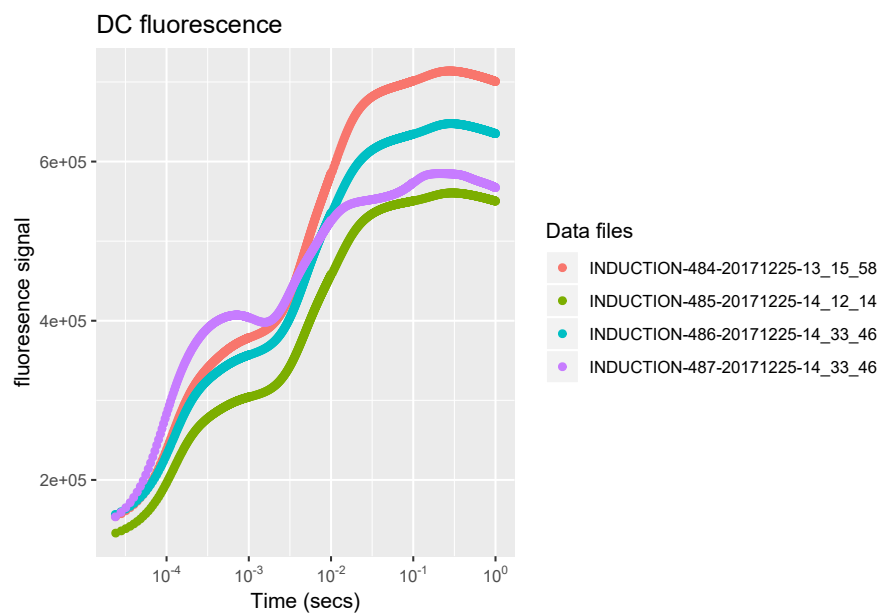


图 9.3: 连续式测量的 ojip 曲线的快速预览

10

环境与配置

本文的内容所完成的 sessioninfo 如下：

```
sessionInfo()
```

```
## R version 3.5.1 (2018-07-02)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 17134)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=Chinese (Simplified)_China.936
## [2] LC_CTYPE=Chinese (Simplified)_China.936
## [3] LC_MONETARY=Chinese (Simplified)_China.936
## [4] LC_NUMERIC=C
## [5] LC_TIME=Chinese (Simplified)_China.936
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods   base
##
## other attached packages:
## [1] jiptest_0.1.3      nls2_0.2          proto_1.0.0
## [4] purrr_0.2.5        ggplot2_3.0.0     minpack.lm_1.2-1
## [7] plantecophys_1.3-2
```

```
##
## loaded via a namespace (and not attached):
## [1] Rcpp_0.12.18      cellranger_1.1.0 compiler_3.5.1  pillar_1.3.0
## [5] highr_0.7         plyr_1.8.4      bindr_0.1.1     MESS_0.5.2
## [9] tools_3.5.1       digest_0.6.17  lattice_0.20-35 evaluate_0.11
## [13] tibble_1.4.2      gtable_0.2.0   pkgconfig_2.0.2 png_0.1-7
## [17] rlang_0.2.2       Matrix_1.2-14  rstudioapi_0.7  yaml_2.2.0
## [21] geeM_0.10.1       xfun_0.4       bindrcpp_0.2.2  withr_2.1.2
## [25] stringr_1.3.1     dplyr_0.7.7    knitr_1.20      tidysselect_0.2.4
## [29] rprojroot_1.3-2   grid_3.5.1     glue_1.3.0      R6_2.2.2
## [33] rematch_1.0.1     geepack_1.2-1  readxl_1.1.0    rmarkdown_1.10
## [37] bookdown_0.7      magrittr_1.5    backports_1.1.2 scales_1.0.0
## [41] htmltools_0.3.6   assertthat_0.2.0 colorspace_1.3-2 labeling_0.3
## [45] stringi_1.2.4     lazyeval_0.2.1 munsell_0.5.0   crayon_1.3.4
```

参考文献

- Ball, J. T., Woodrow, I. E., and Berry, J. A. (1987). *A Model Predicting Stomatal Conductance and its Contribution to the Control of Photosynthesis under Different Environmental Conditions*. Springer Netherlands.
- Baly, E. (1935). The kinetics of photosynthesis. *Proceedings of the Royal Society of London Series B (Biological Sciences)*, (117):218–239.
- Bouvier, A. and Huet, S. (1994). nls2 nonlinear regression by s plus functions. *Computational Statistics & Data Analysis*, 18(1):187–190.
- Buckley, T. N., Martorell, S., Diazspejo, A., Tomàs, M., and Medrano, H. (2014). Is stomatal conductance optimized over both time and space in plant crowns? a field test in grapevine *vitis vinifera*. *Plant Cell & Environment*, 37(12):2707.
- Caemmerer, S. V. (2000). Biochemical models of leaf photosynthesis. *Quarterly Review of Biology*, page 165.
- Cowan, I. R. and Farquhar, G. D. (1977). Stomatal function in relation to leaf metabolism and environment. *Symposia of the Society for Experimental Biology*, 31(23):471.
- De Kauwe, M. G., Lin, Y. S., Wright, I. J., Medlyn, B. E., Crous, K. Y., Ellsworth, D. S., Maire, V., Prentice, I. C., Atkin, O. K., and Rogers, A. (2016). A test of the one point method for estimating maximum carboxylation capacity from field measured, light saturated photosynthesis. *New Phytologist*, 210(3):1130.
- Duursma, R. A. (2015). Plantecophys-an r package for analysing and modelling leaf gas exchange data. *Plos One*, 10(11):e0143346.

- Duursma, R. A. and Medlyn, B. E. (2012). Maespa: a model to study interactions between water limitation, environmental drivers and vegetation function at tree and stand levels, with an example application to CO_2 drought interactions. *Geoscientific Model Development Discussions*, 5(4):919–940.
- Duursma, R. A., Payton, P., Bange, M. P., Broughton, K. J., Smith, R. A., Medlyn, B. E., and Tissue, D. T. (2013). Near-optimal response of instantaneous transpiration efficiency to vapour pressure deficit, temperature and CO_2 in cotton (*Gossypium hirsutum* L.). *Agricultural & Forest Meteorology*, 168(1):168–176.
- Elzhov, T. V., Mullen, K. M., Spiess, A. N., and Bolker, B. (2016). minpack.lm: R interface to the levenberg-marquardt nonlinear least-squares algorithm found in minpack, plus support for bounds.
- Farquhar, G. D., Caemmerer, S. V., and Berry, J. A. (1980). A biochemical model of photosynthetic CO_2 assimilation in leaves of C_3 species. *Planta*, 149(1):78–90.
- Gu, L., Pallardy, S. G., Tu, K., Law, B. E., and Wullschlegel, S. D. (2010). Reliable estimation of biochemical parameters from C_3 leaf photosynthesis-intercellular carbon dioxide response curves. *Plant Cell & Environment*, 33(11):1852–1874.
- Hikosaka, K., Ishikawa, K., Borjigidai, A., Muller, O., and Onoda, Y. (2006). Temperature acclimation of photosynthesis: mechanisms involved in the changes in temperature dependence of photosynthetic rate. *Journal of Experimental Botany*, 57(2):291.
- Huang, H., Dou, X., Sun, P., Deng, B., Wu, G., and Peng, C. (2009). Comparison of photosynthetic characteristics in two ecotypes of *Jatropha curcas* in summer. *Acta Ecologica Sinica*, (29):2861–2867.
- Jones, H. G. (1993). *Plants and microclimate. A quantitative approach to environmental plant physiology*. Cambridge University Press.

- Leuning, R. (1995). A critical appraisal of a combined stomatal-photosynthesis model for C_3 plants. *Plant Cell and Environment*, 18(4):339–355.
- Lobo, F. D. A., Barros, M. P. D., Dalmagro, H. J., Dalmolin, . C., Pereira, W. E., de Souza, . C., Vourlitis, G. L., and Ortíz, C. E. R. (2013). Fitting net photosynthetic light-response curves with microsoft excel a critical look at the models. *Photosynthetica*, 51(3):445–456.
- Medlyn, B. E., Dreyer, E., Ellsworth, D., Forstreuter, M., Harley, P. C., Kirschbaum, M. U. F., Roux, X. L., Montpied, P., Strassmeyer, J., and Walcroft, A. (2002). Temperature response of parameters of a biochemically based model of photosynthesis. ii. a review of experimental data review. *Plant Cell & Environment*, 25(9):1167–1179.
- Medlyn, B. E., Duursma, R. A., Eamus, D., Ellsworth, D. S., Prentice, I. C., Barton, C. V. M., Crous, K. Y., Angelis, P. D., Freeman, M., and Wingate, L. (2011). Reconciling the optimal and empirical approaches to modelling stomatal conductance. *Global Change Biology*, 17(6):2134–2144.
- Prado, C. H. and Moraes, J. A. P. V. D. (1997). Photosynthetic capacity and specific leaf mass in twenty woody species of cerrado vegetation under field conditions. *Photosynthetica*, 33(1):103–112.
- Stinziano, J. R., Mcdermitt, D. K., Lynch, D. J., Saathoff, A. J., Morgan, P. B., and Hanson, D. T. (2018). The rapid a/c_i response: a guide to best practices. *New Phytologist*.
- Stinziano, J. R., Morgan, P. B., Lynch, D. J., Saathoff, A. J., Mcdermitt, D. K., and Hanson, D. T. (2017). The rapid $a-c_i$ response: photosynthesis in the phenomic era. *Plant Cell & Environment*, 40.
- Thornley, J. H. M. (1976). Mathematical models in plant physiology. *London: Academic Press*.
- Tuzet, A., Perrier, A., and Leuning, R. (2003). A coupled model of stom-

atal conductance, photosynthesis and transpiration. *Plant Cell and Environment*, 26(7):1097–1116.

Zhang, X., Shen, S., and Song, J. (2009). The vertical distribution of cotton leaf nitrogen content and photosynthetic characteristics in the north china plain. *Acta Ecologica Sinica*, (29):1893–1898.

ZiPiao, Y. (2010). A review on modeling of responses of photosynthesis to light and co_2 . *Chinese Journal of Plant Ecology*, (06).