

小蓝哥的知识荒原

李详

2021 年 10 月 1 日

简介

我，李详，昵称小蓝哥。现于云南农业大学攻读植物病理学博士学位。

个人博客：[小蓝哥的知识荒原](#)

运营公众号：



1 语言基础

2 统计分析

3 数据可视化

3.1 ggplot2 绘图相关

3.1.1 ggplot2 绘制桑基(冲击图)

```
library(ggfortify)
titanic <- reshape2::melt(Titanic)
# This is how we usually envision data for parallel sets
# head(titanic)
# Reshape for putting the first 4 columns as axes in the plot
titanic <- gather_set_data(titanic, 1:4)
# head(titanic)
# Doing the plotting
ggplot(titanic, aes(x, id = id, split = y, value = value)) +
  geom_parallel_sets(aes(fill = Sex), alpha = 0.3, axis.width = 0.1) +
  geom_parallel_sets_axes(axis.width = 0.2) +
  geom_parallel_sets_labels(colour = 'white', angle = 0, size = 6) +
  scale_y_continuous(expand = c(0, 0)) +
  scale_fill_labeled() +
  theme_minimal() +
  theme(axis.text.y = element_blank(),
        axis.text.x = element_text(size = 12),
        axis.title = element_blank(),
        panel.background = element_blank(),
        panel.grid = element_blank())
```

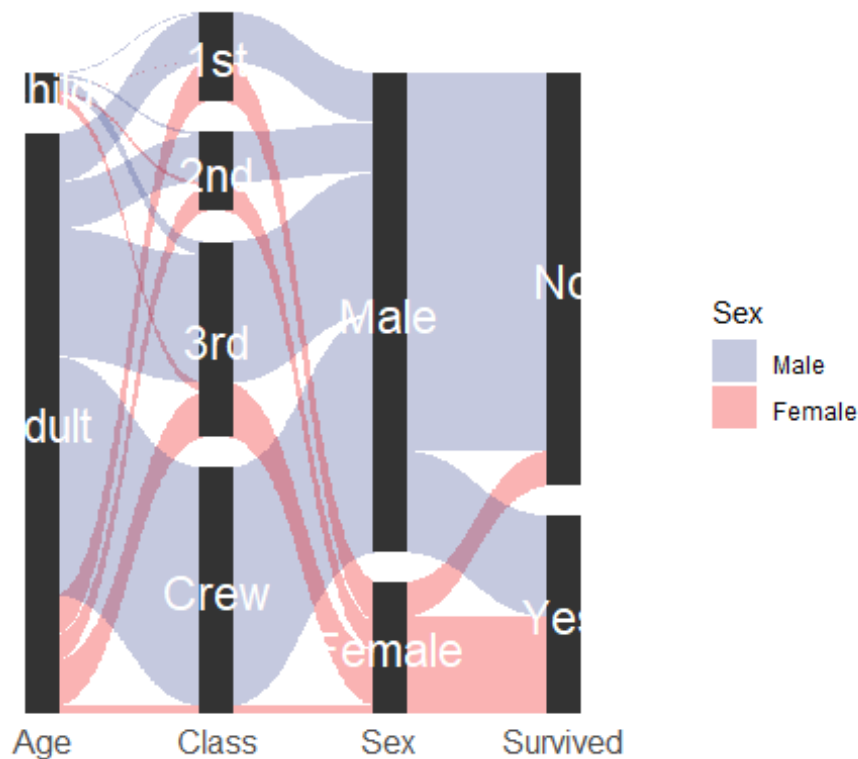


图3.1: ggplot2 绘制桑基图示例

4 R 语言开发

4.1 R 包 pac4xiang 的使用

`pac4xiang` 这个 R 包是我自己开发的，包含了一些我自己常用的函数。包目前在 [GitHub](#) 上，更多功能还在完善中，待进一步开发。

4.1.1 安装

在 R 中运行下面的代码安装 `pac4xiang`。

```
devtools::install_github('https://github.com/lixiang117423/pac4xiang')
```

4.1.2 使用方法

4.1.2.1 calStandCurve

该函数用于计算相对定量 qPCR 的标曲。qPCR 标曲点样参照图 4.1。按照我的思路 and 习惯，我会准备 8 个浓度梯度的 cDNA，从高到低分别是 C0 到 C8。C0 是对应批次下多有样品 cDNA 的组合，然后再以一定的稀释倍数稀释得到 C1，C1 再以同样的倍数稀释得到 C2，依此类推，直到得到 C8。

qPCR标曲点样示意图

		基因1				基因2				基因3			
		技术重复1	技术重复2	技术重复3	技术重复4	技术重复1	技术重复2	技术重复3	技术重复4	技术重复1	技术重复2	技术重复3	技术重复4
		1	2	3	4	5	6	7	8	9	10	11	12
C0	A												
C1	B												
C2	C												
C3	D												
C4	E												
C5	F												
C6	G												
C7	H												

图 4.1: qPCR 标曲点样示意图

函数具体怎么用呢？参考下方的代码：

```
d f . 1 < - c a l S t a n d C u r v e (
  data = "20210927 下机数据 / 20210928 1x_1.txt",
  genes = c("OsPOX223", "OsAPX1", "OsAPX2", "OsPAL"),
  r e p = 3,
  d i l u t i o n = 4,
  s t a r t = 2,
  e n d = 6,
  d r o p . N A = T R U E,
  f i g . t y p e = "p d f"
)
```

其中基因的数量和技术重复的数量乘起来必须是 12；基因的顺序必须是从左到右的；输入稀释倍数后告诉程序后续的标曲建立如何取相应的对数；**start** 表示从第几行开始计算，如 **start = 2** 表示的是从 B 行开始计算标曲；对应的 **end = 6** 表示的是到第 6 行，也就是 F 行结束，也就意味着第一行和最后两行其实是不用的。默认是保存标曲的图片的，如果不需要保存图片加上 **save.fig = FALSE** 即可，默认的图片格式是 pdf 格式，可以自定义。

	Gene	Formula	Slope	Intercept	R2	P.value	max	min	Date
1	OsPOX223	RC = -1.02xCq + 34.22	-1.02	34.22	0.9962	0	29.66	21.87	2021-10-06
2	OsPAL	RC = -0.89xCq + 32.95	-0.89	32.95	0.9991	0	32.73	23.33	2021-10-06
3	OsAPX1	RC = -1.01xCq + 34.94	-1.01	34.94	0.9996	0	30.78	22.60	2021-10-06
4	OsAPX2	RC = -1xCq + 35.11	-1.00	35.11	0.9983	0	31.26	23.05	2021-10-06

图 4.2: 函数 *calStandCurve* 返回的数据框

返回的结果是个数据框（参考图 4.2），包含了基因名称、标曲公式、斜率、截距、 R^2 、P 值、该标曲适用的自变量范围（最大值和最小值）及计算标曲的日期等。保存图片默认存在当前工作目录下（运行 **getcwd()** 查看当前 R 语言工作目录）。

4.1.2.2 calRTqPCR

该函数在 calStandCurve 函数返回的标曲结果的基础上可以计算不同处理下各基因的表达情况。样品的点样方法参考图 4.3。

qPCR样品点样图

			处理1				处理2				处理3			
			基因1	基因2	基因3	基因4	基因1	基因2	基因3	基因4	基因1	基因2	基因3	基因4
			1	2	3	4	5	6	7	8	9	10	11	12
生物学重复1	技术重复1	A												
	技术重复2	B												
生物学重复2	技术重复1	C												
	技术重复2	D												
生物学重复3	技术重复1	E												
	技术重复2	F												
生物学重复4	技术重复1	G												
	技术重复2	H												

图 4.3: qPCR 标曲点样示意图

计算表达量的代码如下：

```
e x p . 1 < - c a l R T q P C R (
data = "20210927 下机数据 / 20210929 1x_1.txt",
StandCurve = df . 1,
genes = c("OsPOX223", "OsAPX1", "OsAPX2", "OsPAL"),
treatment = c("CK", "Inter", "Infect")
)
```

其中的 data 就是样品的下机导出数据，StandCurve 是函数 calStandCurve 返回的数据框，gene 需要严格控制大小写，必须和输入函数 calStandCurve 的基因名称一样，treatment 是指定从左到右的处理分别是什么，这个是为了后续方便进行 t 检验。

4.1.2.3 getAlignResults

该函数可以将软件 CLUSTALW 的输出文件变成.fasta 格式的比对结果。写这个函数的原因在于每次我用 MEGA 进行序列比对构建进化树的时候都报错，不是字符串有问题就是长度不对，索性直接用 Linux 系统下的 CLUSTALW 进行序列比对，然后用 MEGA 构建进化树，百试不爽！

```
my.aln = getAlignResults(aln = 'test.aln')
```

运行完成后会在当前目录下生成一个 YourAlignResults.fasta 文件，就是 fasta 格式的比对文件，然后就可以用来构建进化树了。

- mult.aov 该函数用于多分组数据进行方差分析。
- multGroupTtest 该函数用于进行多分组的 t 检验。这个函数是用于进行多组 t 检验的。比如每次跑 qPCR，会有多个处理多个目的基因，这个时候就需要关注每个目的基因在不同的处理中的表达量。用法参考下方代码：

```
t . t e s t < - m u l t G r o u p T t e s t (
d a t a = e x p . a l l . f i n a l ,
g r o u p 1 = " G e n e " ,
g r o u p 2 = " T r e a t m e n t " ,
C K = " C K " ,
v a l u e = " E x p r e s s i o n " ,
l e v e l = 0 . 9 5
)
```

4.1.2.4 *plot96well*

该函数用于可视化 Roche96 孔板的 Cq 值。

```
plot96well(data = "20210927 下机数据/20210928lx_1.txt")
```

4.1.2.5 *plotCisElement*

该函数用于可视化基因启动子上的顺式作用元件。需要输入的参数有三个：

4.1.2.6 *data*: *plantCARE* 返回的文件;

4.1.2.7 *length*: 启动子长度;

4.1.2.8 *Cis*: 需要展示哪些顺式作用元件。

4.1.2.9 *plotGeneStructure*

该函数用于可视化基因结构。输入文件为 .gtf 文件。 如果同时输入对应的进化树文件，那么就会按照进化树的顺序对基因名称进行排序，可以和 **ggtree** 进行联用。

4.1.2.10 *fasta2df*

这个函数主要是为了将 **fasta** 格式转换成 R 中常见的 **data.frame** 格式。使用很简单，输入 **fasta** 格式即可。

4.1.2.11 *df2fasta*

这个函数主要是将 **data.frame** 格式的序列文件转换成 **fasta** 格式。

4.2 R 包 **ggmotif** 的使用方法

激发我开发这个 R 包的原因是在做基因家族分析中的 **motif** 分析的时候总是找不到趁手的工具去可视化相关的结果，要么是在线网站，要么是别人开发的工具，网页版工具怕哪天就跑路了，别人的工具想要个性化比较困难。那，就自己写一个吧。

R 包下载安装：

```
install.packages("ggmotif")
```

```
# 或 者  
devtools::install_github("lixiang117423/ggmotif")
```

meme 输出的结果主要是 XML 格式和 txt 格式的文件，需要的结果就在这两个文件里面。

4.2.1 从 XML 文件提取结果并绘图

motif 在序列上的位置结果存放在 XML 格式的文件中，运行下面的代码便可以将结果转换成数据框格式并进行可视化。

```
filepath <- system.file("examples", "meme.xml", package="ggmotif")  
motif_extract <- getMotifFromXML(XMLfile = filepath)  
motif_plot <- motif_location(data = motif_extract)  
motif_plot
```

4.2.2 从 txt 文件提取结果并进行可视化

每个 motif 的序列是存放在 txt 文件中的，运行下方的代码便能结果转换成数据框格式并进行绘图。

```
library(tidyverse)  
  
filepath <- system.file("examples", "meme.txt", package = "ggmotif")  
motif_extract <- getMotifFromTxt(data = filepath)  
  
motif_extract %>%  
  dplyr::filter(motif.num == 'Motif.1') %>%  
  dplyr::select(input.seq.motif) %>%  
  ggseqlogo::ggseqlogo() +  
  theme_classic()
```

4.2.3 与 ggtree 联用

有时候会对序列构建进化树展示进化关系，ggmotif 中的函数能够以输入的进化树文件为基础，把输出的 motif 在序列上的位置与进化树上的位置进行匹配，便能够与 ggtree 联用在一个图上展示进化树和 motif 在序列上的位置。

```
filepath <- system.file("examples", "meme.xml", package="ggmotif")  
treepath <- system.file("examples", "tree.nwk", package="ggmotif")  
motif_extract <- getMotifFromXML(XMLfile = filepath)  
motif_plot_with_tree <- motif_with_tree(data = motif_extract, tree = treepath)  
  
p.tree = ape::read.tree(treepath) %>%  
  ggtree::ggtree() +  
  ggtree::geom_tiplabel()  
  
p.tree %>%  
  applot::insert_right(motif_plot_with_tree, width = 0.8)
```

4.3 R shiny 开发

4.3.1 CSS 学习网站

CSS 菜鸟教程。

CSS 在 Rshiny 中的使用。

4.3.2 服务器搭建 Rshiny-server

```
# r o o t 权限进入
sudo su
useradd shiny # 添加 shiny 用户
passwd shiny # 修改 shiny 用户的密码

chmod +w /etc/sudoers # 修改权限
sudo vim /etc/sudoers # 编辑
# root ALL=(ALL) ALL 该行, 在下面添加 shiny ALL=(ALL:ALL) ALL
chmod -w /etc/sudoers # 收回权限

su - shiny # 切换到 shiny 用户

# 安 装 R
sudo apt - get update
sudo apt - get install r-base

# 安 装 s h i n y 包
sudo su -c "R -e \"install.packages('shiny',repos='https://cran.rstudio.
c o m / ' ) \" "

# 安 装 s h i n y s e r v e r
sudo apt - get install gdebi-core
wget https://download3.rstudio.org/ubuntu-14.04/x86_64/shiny-server-1.5.
1 7 . 9 7 3 - a m d 6 4 . d e b
sudo gdebi shiny-server-1.5.17.973-amd64.deb
```

安装 shiny-server 的时候参考官方最新的指导页面，[点击访问](#)。

日志所在目录为：/var/log/shiny-server。需要修改配置文件，保证记录运行日志，否则可能找不到报错提示。

```
vi /etc/shiny-server/shiny-server.conf
preserve_logs true; # 在最上面添加这行
```

软件所在目录为：/srv/shiny-server

安装完成后在浏览器输入 IP:3838 就能访问到示例界面了。默认的端口是 3838 端口，需要其他端口的需要自行调整。

基本管理:

```
# # 重 启
sudo systemctl restart shiny-server
# # 查 看 状 态
sudo systemctl status shiny-server
# # 开 启
sudo systemctl start shiny-server
# # 停 止
sudo systemctl stop shiny-server
```

4.3.3 Ubuntu 安装 devtools

```
sudo apt-get install -y libxml2-dev libcurl4-openssl-dev libssl-dev
```

最重要的是上面这一步，剩下的差啥装啥就 OK。

安装 proj 这个包:

```
sudo apt-get install libproj-dev
install.packages("RcppEigen", INSTALL_opts = '--no-lock')
```

5 Python 基础

5.1 生物序列处理

本章主要是关于 Python 的相关知识，包括 Python 基础知识、数据统计分析、数据可视化及人工智能等内容。

5.1.1 fasta 文件处理

- 处理从 PlantTFDB 下载的序列

```
i m p o r t o s
os.chdir("C://Users//Administrator//Desktop//")
# r e a d t x t m e t h o d o n e
wrky_seq = open('wrky_seq.fasta', 'w')
wrky_info = open('wrky_info.txt', 'w')
f = open("seq.fasta", 'r')
f o r l i n e i n f :
    i f s t r . s t a r t s w i t h ( l i n e , ">" ) :
        seq_name = str.split(line, '|')[0]
        # seq_name = str.split(seq_name, '.')[0]
        seq_species = str.split(line, '|')[1]
        seq_name2 = str.replace(seq_name, ">", "")
        # p r i n t ( s e q _ n a m e )
        wrky_seq.write(seq_name + "\n")
        wrky_info.write(seq_name + " " + seq_name2 + " " + seq_sp
e c i e s + "\n")
e l s e :
    :
```

```
w r k y _ s e q . w r i t e ( l i n e )  
w r k y _ s e q . c l o s e ( )  
wrky_info.close()
```

6 数据统计分析

7 Python 爬虫

8 人工智能

9 基础知识

9.1 学习资源推荐

- 自学生信 RoadMap

9.2 常用的生物学数据库

9.2.1 CATH-Gene3D

这个数据库是蛋白质数据库，主要是蛋白结构域的进化关系。

[点击访问](#)。

9.3 FastTree 构建进化树

FastTree 是基于最大似然法构建进化树的软件，使用简单。Windows 端直接下载，完后运行即可：

```
.\FastTree.exe .\YourAlignResults.fasta
```

前面是软件名称，后面是比对好的序列文件。

10 环境搭建

10.1 WSL 安装使用 Docker

10.1.1 Docker 的安装

参考的安装教程：[Docker-从入门到实践](#)。关键的代码如下：

```
curl -fsSL test.docker.com -o get-docker.sh  
curl -fsSL get.docker.com -o get-docker.sh
```

```
sudo sh get-docker.sh --mirror Aliyun
sudo sh get-docker.sh --mirror AzureChinaCloud
```

10.1.2 Docker 的使用

Docker 默认是需要 root 用户才能使用的，在 Windows 上我习惯于进入 Powershell 后执行下面的命令启动 Docker：

```
w s l      - - s h u t d o w n      #      先 关 闭 w s l
w      s      l      #      启 动 W S L
s u d o      s u      #      进 入 r o o t
s u d o      s e r v i c e      d o c k e r      s t a r t      #      启 动 D o c k e r
su xiang # 切换会用户（非root 权限）
```

10.1.3 如何从 WSL1 切换到 WSL2

我在 Windows 上使用 Docker 遇到的一个很奇怪的问题是，我之前的版本是 WSL1,Docker 无论如何都无法使用，搜索半天也没有找到解决方法，索性将 WSL1 升级成 WSL2，没想到问题就那样解决了。参考教程：[知乎：WSL1 升级为 WSL2](#)。下面是升级的过程：

- 下载对应的内核更新包：[点击下载](#)
- CMD 中管理员身份运行代码：

```
dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /a
ll /norestart
```

- 设置版本

```
wsl --set-version Ubuntu-20.04 2
```

其中的 Ubuntu-20.04 是通过代码 `wsl -l -v` 查看到的。

然后再次重启 WSL 即可。

10.1.4 下载 Docker 镜像

在 [Docker Hub](#) 中检索下载需要的镜像。

10.1.5 Docker 的使用

进入 WSL 后运行下方代码运行 Docker：

```
s u d o      s e r v i c e      - - s t a t u s - a l l
s u d o      s e r v i c e      d o c k e r      s t a r t
s u d o      s e r v i c e      d o c k e r      s t a r t
docker run -v /mnt/./work -it omicsclass/rnaseq
```

其中的 work 是不一定的，需要看镜像给的路径是啥。

10.1.6 如何创建自己的镜像

先从 [Docker Hub](#) 下载 Ubuntu 的官方镜像，然后在镜像中安装需要的软件。 PS：如何加速 pip 的下载：

```
pip install django -i https://pypi.tuna.tsinghua.edu.cn/simple
```

加速的 R 包的下载安装：

```
options(repos=structure(c(CRAN="https://mirrors.tuna.tsinghua.edu.cn/CRA
N/"),
options(Bioc_mirror="https://mirrors.tuna.tsinghua.edu.cn/bioconductor"))
```

在安装完成需要的软件后，先运行 `exit` 退出 Docker，然后运行下面的代码生成新的镜像：

```
docker commit -m " add some softwares for RNA-Seq" -a "xiangli" 21bfa81
0c811 lixiang117423/rnaseq:v1
```

然后登陆自己的 Docker，登录以后把新的镜像推送到 Docker Hub 即可：

```
docker push lixiang117423/rnaseq:v1
```

10.2 Conda 的安装使用

10.2.1 下载安装

现在[官方网站](#)下载对应版本的.sh 文件。然后一路默认安装即可。安装完成后激活用户目录下的.bashrc 文件即可。

10.2.2 Conda 安装 R 及 R 包

```
c o n d a i n s t l l r - b a s e
c o n d a i n s t l l r - g g p l o t 2
conda install -c bioconda bioconductor-deseq2
```

10.2.3 安装其他软件

其他软件的安装直接去 [Conda Gallery](#) 上检索进行安装。

10.2.4 pip 下载速度慢的解决方法：

```
pip install django -i https://pypi.tuna.tsinghua.edu.cn/simple
```

10.3 git 的使用

10.3.1 git 同时管理 GitHub 与 Gitee

```
1 . 查看 远 程 仓 库 地 址
g i t r e m o t e - v

2 . 重 命 名 远 程 仓 库
g i t r e m o t e r e n a m e o r i g i n g i t h u b
```

3 . 添加新的远程仓库
git remote add gitee https://gitee.com/gozhuyinglong/blog-demos.git

4 . 多个远程仓库推拉
git push git hub main
git pull git hub main
git push git ee main
git pull git ee main

5 . 将本地分支与远程分支进行关联
git push --set-upstream gitee main

6 . 直接推送不指定分支
git push git hub
git pull git hub
git push git ee
git pull git ee

7 . 移除远程仓库
git remote remove gitee

10.4 软件安装

10.4.1 Aspera

```
wget https://ak-delivery04-mul.dhe.ibm.com/sar/CMA/OSA/09cne/0/ibm-aspera-connect-3.11.0.5-linux-g2.12-64.tar.gz
```

```
tar -zxvf ibm-aspera-connect-3.11.0.5-linux-g2.12-64.tar.gz
```

```
./ibm-aspera-connect-3.11.0.5-linux-g2.12-64.sh
```

```
echo ""export PATH=\"/home/xiang/.aspera/connect/bin:~> ~
```

```
so ur ce ~ / . b a s h r c
```

```
a s c p - h
```

```
ascp-i /mnt/c/Users/Administrator/Desktop/aspera.openssh -QT -l100m -k1  
d /mnt/h/BaiduNetdiskDownload/*.gz subasp@upload.ncbi.nlm.nih.gov:uplo  
ads/luolifenllf_126.com_tCdfcqJq/20210816upload
```

```
ascp -i aspera.openssh -QT -l1000m -k1 -d A2.R2.fastq.gz subasp@upload.  
ncbi.nlm.nih.gov:uploads/luolifenllf_126.com_tCdfcqJq/20210816upload
```

10.5 Ubuntu 搭建生信环境

下面的代码来自王通老师。

10.5.1 旧版本

```
# 下 载 u b u n t u 2 0 . 0 4 镜 像
https://ftp.sjtu.edu.cn/ubuntu-cd/20.04.1/ubuntu-20.04.1-desktop-amd64.
i s o

# 制 作 U S B 启 动 盘
# 下 载 使 用 r u f u s https://rufus.ie/
# 或 者 u l t r a I S O https://cn.ultraiso.net/

# 挂 载 磁 盘
# 假 设 未 挂 载 磁 盘 为 / d e v / s d b
f d i s k / d e v / s d b
p a r t e d / d e v / s d b
# 交 互 界 面
m k l a b e l i g p t
q u i t
# 格 式 化 磁 盘
m k f s . x f s - f / d e v / s d b
# 挂 载 磁 盘
m k d i r / i f s 1
m o u n t / d e v / s d b / i f s 1
# 修 改 f s t a b
echo "/dev/sdb /ifs1 xfs defaults,uquota
0 0 " > / e t c / f s t a b

# 创 建 r o o t 账 户
s u d o p a s s w d r o o t
输 入 当 前 用 户 密 码 :
输 入 r o o t 密 码 :
再 次 输 入 r o o t 密 码 :

# 以 下 操 作 使 用 r o o t 账 户 完 成
s u

# 修 改 源
c p / e t c / a p t / s o u r c e s . l i s t / e t c / a p t / s o u r c e s . l i s t . b a k
s e d - i ' s # c n . a r c h i v e . u b u n t u . c o m # m i r r o r s . a l i y u n . c o m # g ' / e t c / a p t / s o u r c e s .
l i s t
s u d o a p t - g e t u p d a t e && s u d o a p t - g e t u p g r a d e

# s s h 登 录
a p t i n s t a l l o p e n s s h - s e r v e r
p s - e | g r e p s s h # 检 查 s s h s e r v e r 是 否 启 动
```

```
sudo /etc/init.d/ssh start # 启动
```

```
# 安装 java
apt-get install openjdk-8-jdk
sudo add-apt-repository ppa:webupd8team/java
apt-get update
apt install -y oracle-java8-installer
apt install -y zlib1g zlib1g-dev
apt install -y libboost-dev
```

```
# 安装 R 以及 R studio
apt install -y r-base
https://download1.rstudio.org/desktop/bionic/amd64/rstudio-1.3.1093-amd64.deb
dpkg -i rstudio-1.3.1093-amd64.deb
```

```
# 安装一些小工具
apt install -y git vim tree cren htop cmake lftp lrzsz
```

```
# apt 安装生物软件
apt install -y bwa
apt install -y samtools
apt install -y bcftools
apt install -y blast2
apt install -y bedtools
apt install -y seqtk
apt install -y minimap2
apt install -y bowtie2
apt install -y phylip
apt install -y clustalx
apt install -y canu
apt install -y kraken2
apt install -y hisat2
apt install -y stringtie
apt install -y jellyfish
apt install -y circos
apt install -y nanopolish
apt install -y nanoawk
apt install -y centrifuge
apt install -y rna-star
apt install -y freebayes
apt install -y cnvkit
apt install -y spades
apt install -y motthur
apt install -y muscle
apt install -y mafft
apt install -y iqtree
apt install -y sniffles
apt install -y last-align
```

a p t i n s t a l l - y a u g u s t u s
a p t i n s t a l l - y b a m t o o l s
a p t i n s t a l l - y b e d o p s
a p t i n s t a l l - y d e l l y

```
# 安 装 浏 览 器
apt-get install -y chromium-browser
```

```
# 安 装 微 信
wget https://www.ubuntukylin.com/public/pdf/wine-wechat_1.0-windows2.8.6_a11.d_eb
```

```
# Debian 系的操作系统可以执行这样的命令：
sudo dpkg --add-architecture i386
sudo apt update
# 安装 wine 应用所需的依赖，也就是 wine：
sudo apt install wine-stable
# 使用 dpkg 安装 wine 应用：
sudo dpkg -i wine-wechat_1.0-windows2.8.6_all.deb
```

```
# 安 装 b i o c o n d a
wget https://repo.continuum.io/miniconda/Miniconda3-latest-Linux-x86_64.
sh
sh Miniconda3-latest-Linux-x86_64.sh
source ~/bin/bashrc
```

```
# 添 加 软 件 源
conda config --add channels bioconda
conda config --add channels conda-forge
```

10.5.2 新版本

[illegible]

制作 USB 启动盘
下载使用 [rufus https://rufus.ie/](https://rufus.ie/)
或者 [ultraISO https://cn.ultraiso.net/](https://cn.ultraiso.net/)

#			挂			载			磁				盘
#	假	设	未	挂	载	磁	盘	为	/	d	e	v	/s d b
f		d		i		s		k				-	l
p	a	r	t	e	d		/	d	e	v	/	s	d b
#			交			互			界			面	
m	k		l	a		b	e	l				g	p t
q				u					i				t


```
# 格式化磁盘
mkfs.xfs -f /dev/sdb
# 挂载
mkdir /ifs1
mount /dev/sdb /ifs1
# 修改fs type
echo "/dev/sdb /ifs1 xfs defaults,uquota
0 0 " > /etc/fstab
```

如果不安装系统，购买腾讯云服务器一年 74 元，2 核心 4G 版本即可
<https://curl.qcloud.com/gm6m0QoY>

```
#####
# 基本配置 #
#####
# 创建root账户 #
sudo passwd root
# 输入当前用户密码:
# 再次输入root密码:
# 再次输入root密码:
```

```
# 以下操作使用root账户完成
su -
```

```
# 修改源
cp /etc/apt/sources.list /etc/apt/sources.list.bak
sed -i 's#cn.archive.ubuntu.com#mirrors.aliyun.com#g' /etc/apt/sources.
list
sudo apt-get update && sudo apt-get upgrade
```

```
# ssh 登录
apt install openssh-server
ps -e | grep ssh # 检查ssh server 是否启动
sudo /etc/init.d/ssh start # 启动
```

```
# 安装jdk
apt-get install openjdk-8-jdk
sudo add-apt-repository ppa:webupd8team/java
apt-get update
apt install -y oracle-java8-installer
apt install -y zlib1g zlib1g-dev
apt install -y libboost-dev
```

```
#####
# 安装R以及Rstudio #
#####
apt install -y r-base
```

```
https://download1.rstudio.org/desktop/bionic/amd64/rstudio-1.3.1093-amd64.pkg -i rstudio - 1 . 3 . 1 0 9 3 - a m d 6 4 . d e b
```

```
# 安 装 一 些 小 工 具  
apt install -y git vim tree cren htop cmake lftp lrzsz
```

```
# a p t 安 装 生 物 软 件  
a p t i n s t a l l - y b w a  
a p t i n s t a l l - y s a m t o o l s  
a p t i n s t a l l - y b c f t o o l s  
a p t i n s t a l l - y b l a s t 2  
a p t i n s t a l l - y b e d t o o l s  
a p t i n s t a l l - y s e q t k  
a p t i n s t a l l - y m i n i m a p 2  
a p t i n s t a l l - y b o w t i e 2  
a p t i n s t a l l - y p h y l i p  
a p t i n s t a l l - y c l u s t a l x  
a p t i n s t a l l - y c a n u  
a p t i n s t a l l - y k r a k e n 2  
a p t i n s t a l l - y h i s a t 2  
a p t i n s t a l l - y s t r i n g t i e  
a p t i n s t a l l - y j e l l y f i s h  
a p t i n s t a l l - y c i r c o s  
a p t i n s t a l l - y n a n o p o l i s h  
a p t i n s t a l l - y n a n o o k  
a p t i n s t a l l - y c e n t r i f u g e  
a p t i n s t a l l - y r n a - s t a r  
a p t i n s t a l l - y f r e e b a y e s  
a p t i n s t a l l - y c n v k i t  
a p t i n s t a l l - y s p a d e s  
a p t i n s t a l l - y m o t h u r  
a p t i n s t a l l - y m u s c l e  
a p t i n s t a l l - y m a f f t  
a p t i n s t a l l - y i q t r e e  
a p t i n s t a l l - y s n i f f l e s  
a p t i n s t a l l - y l a s t - a l i g n  
a p t i n s t a l l - y a u g u s t u s  
a p t i n s t a l l - y b a m t o o l s  
a p t i n s t a l l - y b e d o p s  
a p t i n s t a l l - y d e l l y
```

```
# 安 装 浏 览 器  
apt - get install -y chromium - browser
```

```
# 安 装 微 信  
wget https://www.ubuntukylin.com/public/pdf/wine-wechat_1.0-windows2.8.6 - a l l . d e b
```

```
# D e b i a n 系 的 操 作 系 统 可 以 执 行 这 样 的 命 令 :
```

```
sudo dpkg --add-architecture i386
sudo apt update
# 安装 wine 应用所需的依赖, 也就是 wine:
sudo apt install wine-stable
# 使用 dpkg 安装 wine 应用:
sudo dpkg -i wine-webkit_1.0-windows2.8.6_all.deb
```

```
#####
#          安装 bioconda          #
#####
wget https://repo.continuum.io/miniconda/Miniconda3-latest-Linux-x86_64.
sh
sh Miniconda3-latest-Linux-x86_64.sh
source ~/.bashrc
```

```
#          添          加          软          件          源
conda config --add channels bioconda
conda config --add channels conda-forge
```

```
#####
# 使用 bioconda 安装常用软件      #
#####
#          安          装          m          a          m          b          a
conda install -c conda-forge -y mamba
```

```
# 利 用 m a m b a 安 装 软 件
mamba install -y bwa
mamba install -y samtools
mamba install -y bcftools
mamba install -y blast
mamba install -y blat
mamba install -y mummer
mamba install -y mafft
mamba install -y muscle
mamba install -y lastz
mamba install -y sratools
mamba install -y seqkit
mamba install -y seqtk
mamba install -y bedtools
mamba install -y bedops
mamba install -y gfastools
mamba install -y circos
mamba install -y entrez-direct
mamba install -y emboss
```

```
#          安          装          数          据          质          控          软          件
mamba install -y fastqc multiqc
mamba install -y trimomatic
```

```

m a m b a i n s t a l l - y f a s t p
# 安 装 基 因 组 拼 接 相 关 工 具
m a m b a i n s t a l l - y v e l v e t
m a m b a i n s t a l l - y f l y e
m a m b a i n s t a l l - y m i n i a s m
m a m b a i n s t a l l - y c a n u
m a m b a i n s t a l l - y m e g a h i t
m a m b a i n s t a l l - y s p a d e s
m a m b a i n s t a l l - y q u a s t
m a m b a i n s t a l l - y r a c o n
m a m b a i n s t a l l - y m i n i a s m
m a m b a i n s t a l l - y n a n o p o l i s h

# 安 装 基 因 功 能 分 析 软 件
m a m b a i n s t a l l - y p r o d i g a l
m a m b a i n s t a l l - y g l i m m e r
m a m b a i n s t a l l - y a u g u s t u s
m a m b a i n s t a l l - y t r f

#####
# 使用 虚 拟 环 境 #
#####
# 1 创 建 p y t h o n 2 . 7 环 境
conda create -n py27 -y python=2.7
# 查 看 现 有 虚 拟 环 境
conda env list
# 激 活 p y t h o n 2 . 7 环 境
conda activate py27
# 查 看 p y t h o n 版 本
python -V

# 2 安 装 n a n o p l o t
conda create -n nanoplot -y nanoplot

# 3 b i o c o n d a 管 理 R 及 R 包
mamba install -y r-base=4.1.1
mamba install -y bioconductor-deseq2
mamba install -y bioconductor-rnaseqgene
mamba install -y bioconductor-clusterProfiler

```

11 基因组学

11.1 基因家族分析

基因家族分析是一种常见的生物信息学分析套路，也是生物信息学数据挖掘发表小文章常用的分析方法，和 GEO 挖掘等类似。基因家族的分析鉴定可以用 [pfam](#) 上

的 **hmm** 文件进行基因家族的检索鉴定，也可以用 **blast** 的方法进行比对鉴定，通常是适用拟南芥对应的基因家族进行比对鉴定。植物转录因子数据库，[点击访问](#)。

11.1.1 基因家族分析思路及文章撰写思路

基因家族分析是继 GEO 数据挖掘后，一种新的生物信息学挖掘策略。

如何做基因家族研究，可以参考这个帖子：
<http://www.planttech.com.cn/blog/58882464a46>。

为什么我们要选择三七呢？

首先三七参考基因组还不是很多，我们实验室有一个，今年杨生超副校长有一篇最新的，这些基因组数据为我们后续做三七基因家族研究提供了坚实的基础；其次，我们实验室以三七为主，有大量的三七资源，方便我们后续验证。

文章撰写的格式建议以 **Frontiers in Plant Science** 为模板，后续方便修改。

每个基因家族都有大量的综述，因为基因家族基本都是转录因子，转录因子基本都是有很好的综述的。

11.1.2 数据准备

需要准备的数据主要是参考基因组数据，包括 **fasta** 格式的序列文件、**gff** 或 **gtf** 格式的基因组注释文件、蛋白质序列文件（通常是每个转录本的蛋白序列）、**cDNA** 序列等文件。如果有转录组数据的话进行对应的转录组分析即可。除开上述这些文件外，还需要适用的文件还有 **.hmm** 格式的文件。

11.1.3 软件准备

只需要会使用 **Linux** 系统，会安装 **Docker** 即可，然后下载组学大课堂的镜像即可。[点击浏览](#)镜像地址。**Docker** 的安装适用方法参考 10.1。

11.1.4 分析过程

- **mRNA_ID** 与基因 ID 的提取 由于一个基因会对应多个转录本，因此一个基因下会对应多个 **mRNA** 的编号。在后续的分析中，每个基因只需要选择一个转录本的编号进行分析即可，因为每个基因不同的转录本的序列差异是较小的。提取的代码：

```
perl code/script/mRNAid_to_geneid.pl data/unzip_data/*.gff* results/step.1.get.mRNA.and.gene.ID/mRNA2geneID.txt
perl code/script/geneid_to_mRNAid.pl data/unzip_data/*.gff* results/step.1.get.mRNA.and.gene.ID/geneID2mRNAid.txt
```

- **检索结构域** 这一步主要是以 **.hmm** 文件为基础检索该物种蛋白序列中含有该结构域的序列。输入文件包括 **.hmm** 文件和蛋白文件，输出 **hmmsearch** 的检索结果。其中用于后续筛选的是 **evaluate** 这个参数，部分文章以 0.001 为阈值。**of** 那一列表示的是某个基因对应的这个结构域有几个。

```
hmmsearch --domtblout results/step.2.domain.search/hmm.txt --cut_tc data/unzip_data/*.hmm data/unzip_data/*.pep*
```

- 选择结构域 由于一个基因的单个转录本可能会比对到多个结构域，因此需要对比对到的结构域进行选择。默认选择的是第一个结构域。下面代码的最后一个参数是 `hmmsearch` 输出文件里面的 E-value,如果需要全部的第一个结构域，将阈值设置为 1 即可。

```
perl code/script/domain_xulie.pl results/step.2.domain.search/hmm.txt data/unzip_data/*.pep* results/step.2.domain.search/domain.fa 1.2e-28
```

- 多序列比对 之所以要进行多序列比对，是因为下载的.hmm 文件是来自很多物种的这个结构域组成的隐马尔科夫模型，进行多序列比对后将该物种检索到的结构域序列进行比对，再次构建该物种该基因家族的隐马尔科夫模型，会更加准确。

```
echo -e '1\nresults/step.2.domain.search/domain.fa\n2\n1\nresults/step.2.domain.search/out.aln\nnr.domain.search/out.dnd\nX\n\n\nX\n' |clustalw
```

- 重构隐马尔科夫模型

```
hmmbuild results/step.2.domain.search/new.hmm results/step.2.domain.search/out.aln
```

- 重新进行检索 利用构建得到的新的隐马尔科夫模型重新进行检索结构域。

```
hmmsearch --domtblout results/step.2.domain.search/new.out.txt --cut_tc results/step.2.domain.search/new.hmm data/unzip_data/*.pep*
```

- 筛选输出结果 对重新检索后的结果进行筛选，也是对 E-value 进行筛选。

```
grep -v "^#" results/step.2.domain.search/new.out.txt|awk '$7<0.001 {print}' > results/step.2.domain.search/domain.new.out.selected.txt
```

- 去除重复的 ID 上一步筛选得到的是该种中哪些基因是潜在的目标基因家族成员，而一个基因对应了多个 mRNA，因此，只需要在筛选后的每个基因中选择一个具有代表性的 mRNA 进行后续的分析即可。这个提取唯一 ID 的步骤需要手动完成（PS：手动完成也很快）。手动挑选完 mRNA 的 ID 放在第一列，另存为文件 `uniqueID.txt`。

```
perl code/script/select_redundant_mRNA.pl results/step.1.get.mRNA.and.gene.ID/mRNA2geneID.txt results/step.2.domain.search/domain.new.out.selected.txt results/step.2.domain.search/remove_redundant_IDlist.txt
```

- 提取蛋白序列 在得到基因 ID 后需要提取蛋白序列进行后续的分析。在 SMART 或者 Pfam 或 NCBI CDD 确认这些基因是真真正正含有该结构域，没有的基因要剔除！在 SMART 中没有检索到结构域的基因在 `gene.not.in.SMART.txt` 中；在 Pfam 中全都是 WRKY 结构域，对应文件为 `Pfam.results.txt`。

```
perl code/script/get_fa_by_id.pl results/step.2.domain.search/uniqueID.txt data/unzip_data/*.pep* results/step.2.domain.search/pep.needs.confirm.fa
```

- 计算蛋白质分子量等

```
perl code/script/stat_protein_fa.pl results/step.2.domain.search/pep.confirmed.fa results/step.2.domain.search/pep.MW.txt
```

- 构建进化树 选择利用软件 CLUSTALW 进行多序列比对，然后利用 MEGA 构建进化树。CLUSTALW 输出结果转换成.fasta 格式的方法参考 4.1。
- Motif 分析

```
meme results/step.3.seq.and.tree/pep_confirmed.fa -protein -oc results/step.4.motif/ -nostatus -time 18000 -maxsize 6000000 -mod anr -nmotifs 10 -minw 6 -maxw 100
```

- 基因结构分析

```
perl code/script/get_gene_exon_from_gff.pl -in1 results/step.2.domain.search/uniqueID.txt -in2 data/unzip_data/*.gff* -out results/step.5.gene.structure/gene_exon_info.gff
```

- 基因在染色体上的定位

```
samtools faidx data/unzip_data/*.dna* cp data/unzip_data/*.fai results/step.5.gene.structure/ perl code/script/get_gene_weizhi.pl -in1 results/step.2.domain.search/uniqueID.txt -in2 data/unzip_data/*.gff* -out results/step.5.gene.structure/mrna_location.txt
```

- 顺式作用元件分析 脚本默认的启动子长度是 1500bp。将提取得到的启动子序列上传到 [Plane CARE](#) 进行分析。

```
perl code/script/get_promoter.pl data/unzip_data/*.dna.top* results/step.5.gene.structure/mrna_location.txt results/step.6.cis.acting.element/promoter.txt
```

- 基因家族成员的亚细胞定位分析 两个网站: [WolfPsort](#) 和 [Cello](#)。

12 RNA-Seq

13 代谢组学

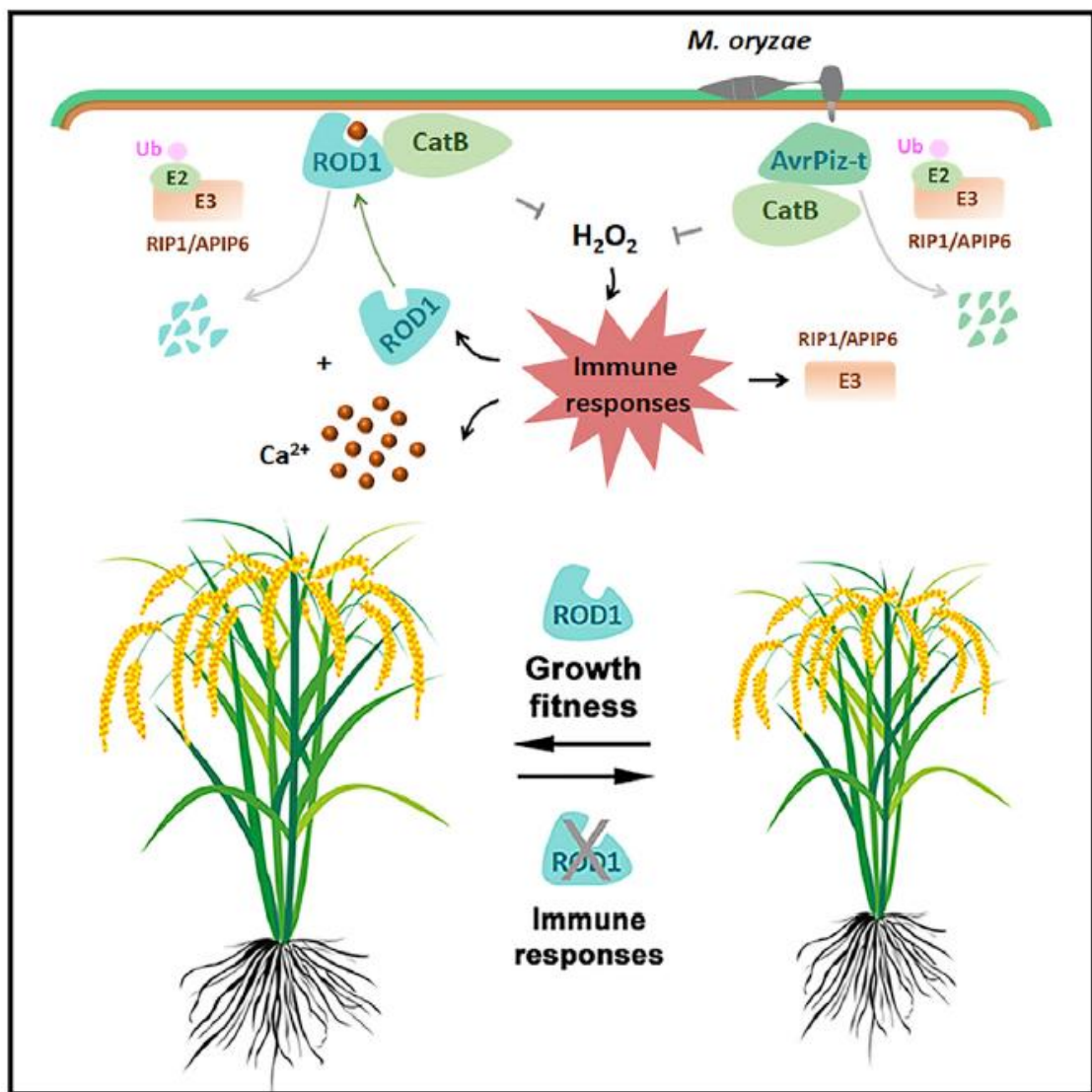
14 蛋白质组

15 多组学

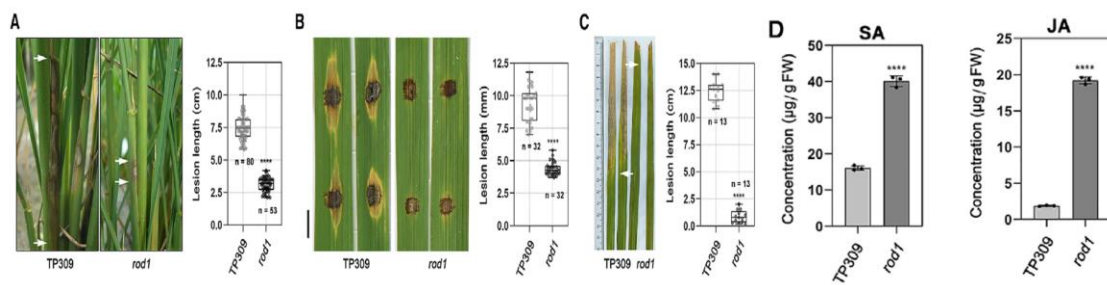
16 植物病理学

16.1 何祖华 2021*Cell* 文章

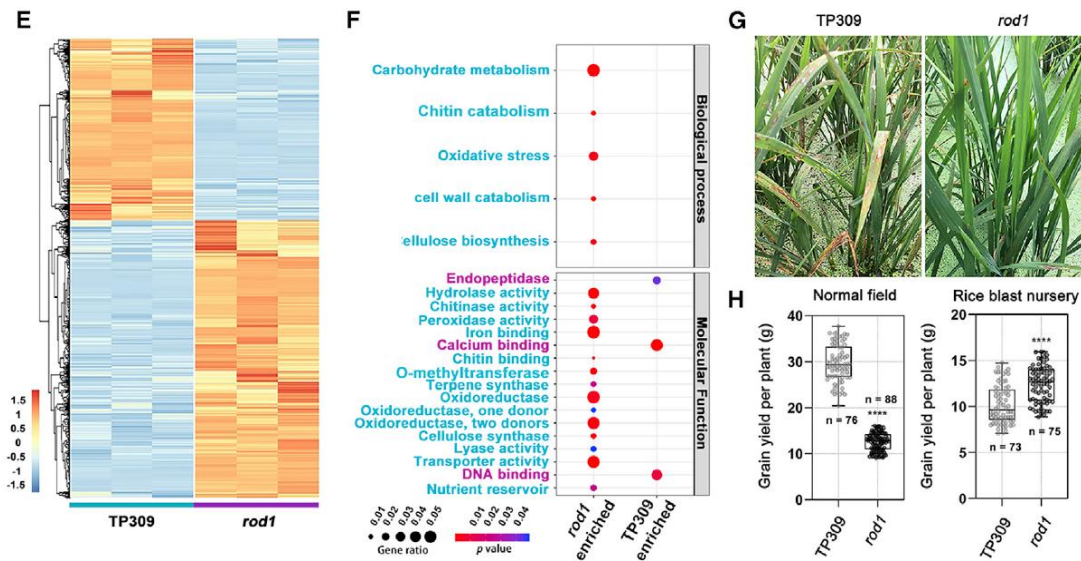
这是一篇让人读完以后叹为观止的文章([Gao et al. 2021](#)), 精彩, 实在是精彩!



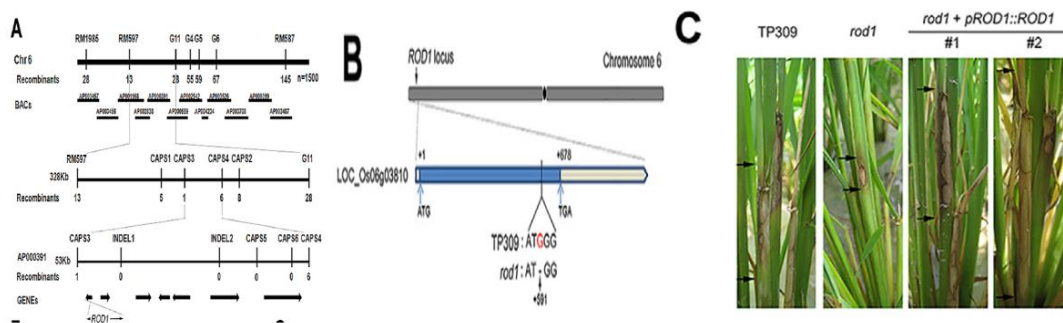
故事是这样开始的，在 TP309（粳稻）育种群体中，发现一个很特殊的变种，这个变种携带了一个叫做 *rod1* 的基因，这个基因有什么特别的地方呢？通过田间试验和室内试验发现，携带了 *rod1* 基因的植株对水稻纹枯病、水稻稻瘟病和水稻白叶枯病这三种严重影响水稻生产的病害都具有广谱抗性（下图 ABC）；而且，其植株体内的 SA 和 JA 的含量也比 TP309 的高很多（下图 D）。



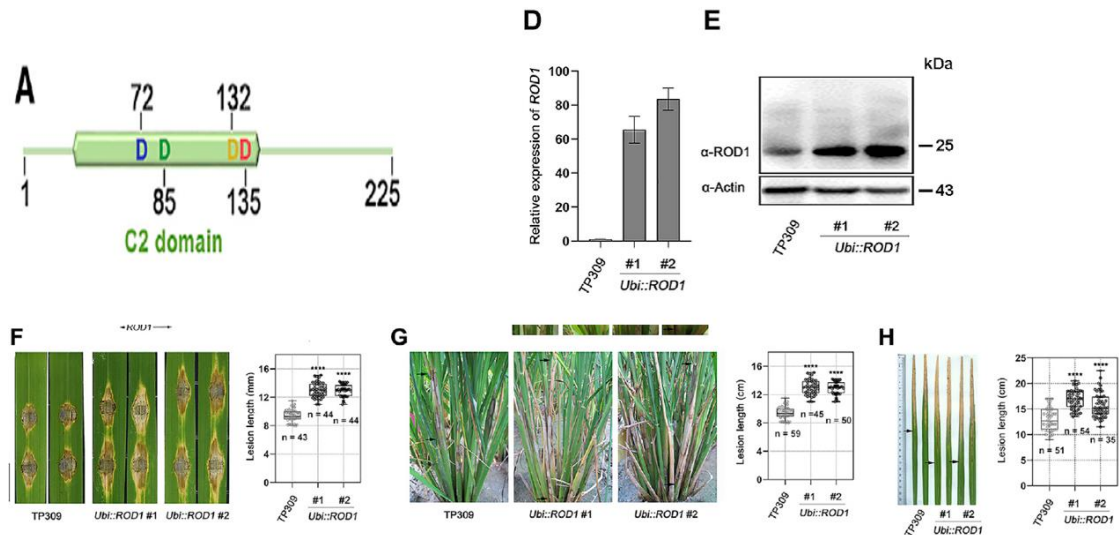
为了验证 *rod1* 的广谱抗性，利用常规的 RNA-Seq 对 TP309 和携带 *rod1* 的植株进行转录组测序，测序的结果可以说是非常明显了，但从热图上看是完全分开的（下图 E），然后呢对这些差异基因进行 GO 富集分析，发现有几个很特殊的 GO term 在 *rod1* 里面没有，这几个通路分别是内肽酶、钙结合及 DNA 结合（下图 F 紫色的字）。*rod1* 中显著富集到的 GO term 有离子结合、氧化酶等。于是，作者他们认为 *rod1* 中 Ca^{2+} 与 ROS 两者之间存在某种关系，导致其免疫上调。然后在田间观察 TP309 和 *rod1* 对水稻产量的影响，发现在自然田间条件下的话，*rod1* 植株的单株产量远低于 TP309 的；但是在天然的稻瘟病苗圃中，*rod1* 的单株产量远高于 TP309。这个结果能说明啥呢？这个结果说明如果有病害侵染的话，*rod1* 植株会表现得更好，也就是产量更高。



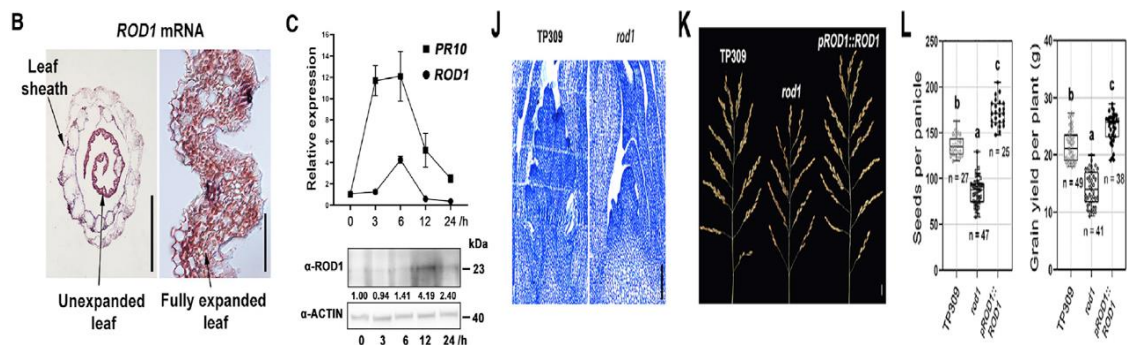
下一步要做的，肯定是把这个基因克隆出来，只有克隆出来，才能做进一步的深入研究。利用图位克隆技术在水稻的 6 号染色体附近大概 53kb 的一个区域定位到该基因（下图 A），命名为 *ROD1*；进一步比较了水稻参考基因组水稻品种 Nipponbare、TP309 及 *rod1* 在该区域的序列差异，发现 *rod1* 存在一个核苷酸的缺失，也就是 TP309 里面的是 ATGGG，在 *rod1* 里面是 AT-GG，中间少了个 G（下图 B）。为了确定他们鉴定的这个基因，利用遗传互补（也就是利用野生型基因补偿突变型基因使其恢复正常的表型）确认了该基因的功能（下图 C）。



这个基因的蛋白长度大概是 225 个氨基酸长度，并且含有一个 C2 结构域（蛋白质激酶 C 保守结构域 2，承担信号转导和细胞膜转运功能），这个结构域上还含有 4 个天冬氨酸残基（下图 A）。然后利用玉米泛素启动子将 *ROD1* 过表达，过表达以后呢制作对水稻上的三种病害的抗性都降低了（下图 DEFGH）。这个结果说明 *ROD1* 是水稻免疫抑制子，也就是它越表达，水稻的抗性越低。

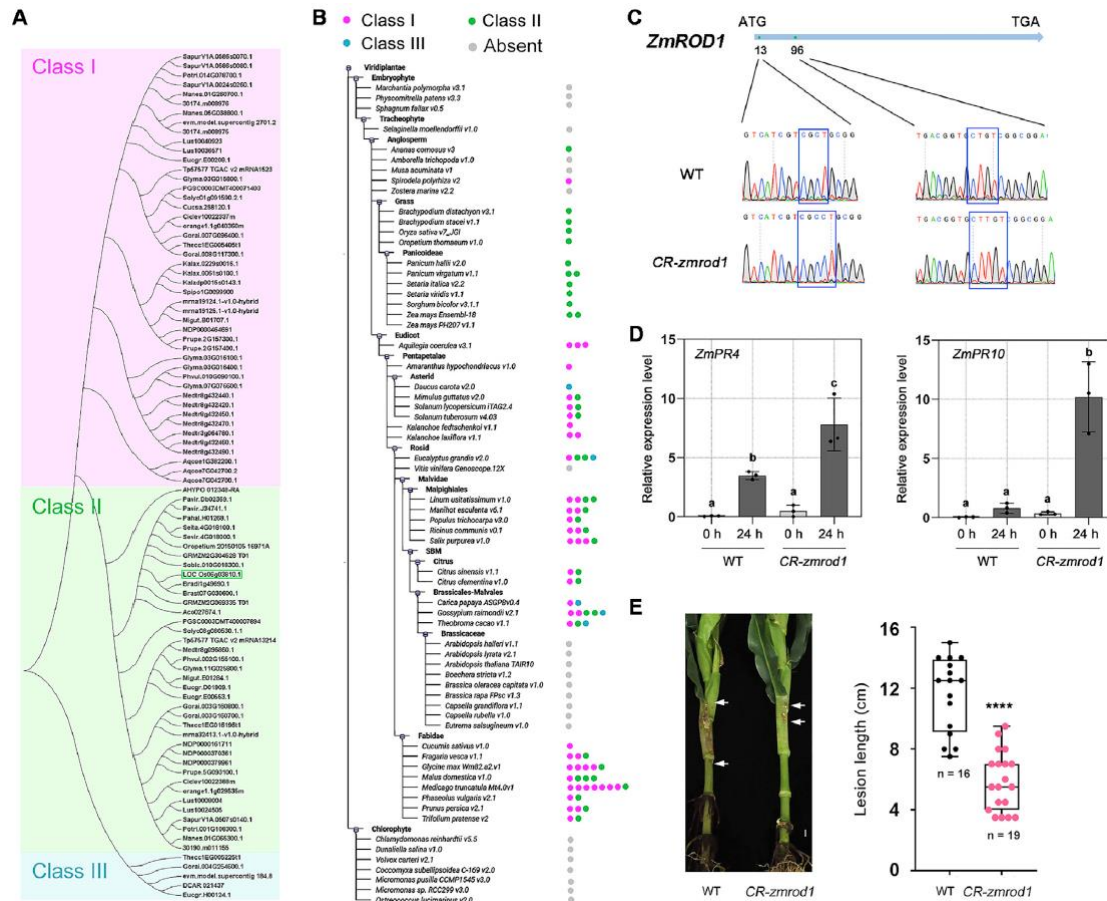


不同的基因在植株体内表达的位点是不一样的，有的基因在根系表达多，有的基因在叶片中表达更旺盛。那 *ROD1* 主要是水稻植株的那个部位表达呢？通过原位杂交（*In situ hybridization*），发现 *ROD1* 主要是在叶片上进行表达（下图 B），而叶片更好是更多病原菌侵染的部位。而且在水稻植株被水稻白叶枯病菌侵染后，*ROD1* 的表达会被诱导上调（下图 C）。*rod1* 在田间生长的时候穗子比 TP309 短，穗子上的分支也更少（下图 K），但是在过表达 *ROD1* 后，穗子更长，产量也更高（下图 KL）。这个结果表明 *ROD1* 在水稻的抗性和生长适应性两方面具有不同的作用。也就是 *rod1* 的抗性高，但是产量低；*ROD1* 的抗性低，但是产量高。这也符合植物生长-免疫平衡的基本原则。



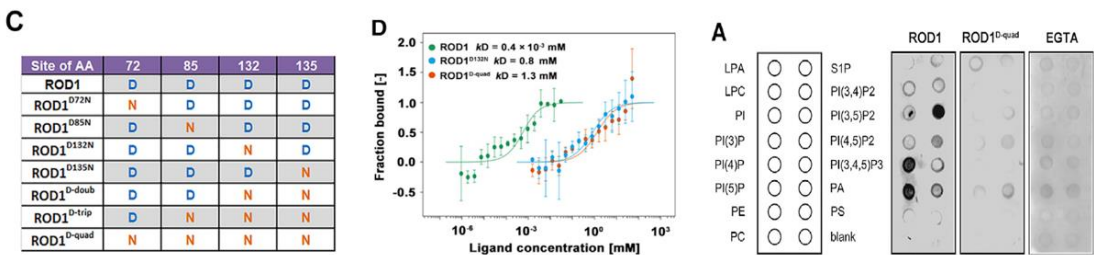
再下一步是探究 *ROD1* 的同系物的分布情况，在开花植物中都有其同系物的存在，但是在苔藓等低等植物中没有其同系物的踪迹（下图 AB）。这个结果表明 *ROD1* 这一类 C2 结构域蛋白出现在植物进化的较晚期。*ROD1* 类蛋白在单子叶作物中有

着极高的序列保守性，但是在拟南芥等十字花科的植物中却有着较高的多样性（下图 B）。要是我分析，肯定是到这里就结束了，但是作者团队在这个地方提出的问题是“**ROD1** 类蛋白在谷物类作物的免疫过程中是否有保守作用呢？”。于是呢，就用 **CRISPR/Cas9** 技术在玉米上搞了个突变体，结果发现突变体中 **PR** 基因的表达量更高，而且突变体对 *R.solani* 的抗性更强（下图 CDE）。这个结果加上前面的结果表明 **ROD1** 及其同系物在谷物中是一类特别的感病基因，在谷物免疫的过程中扮演者保守的功能。

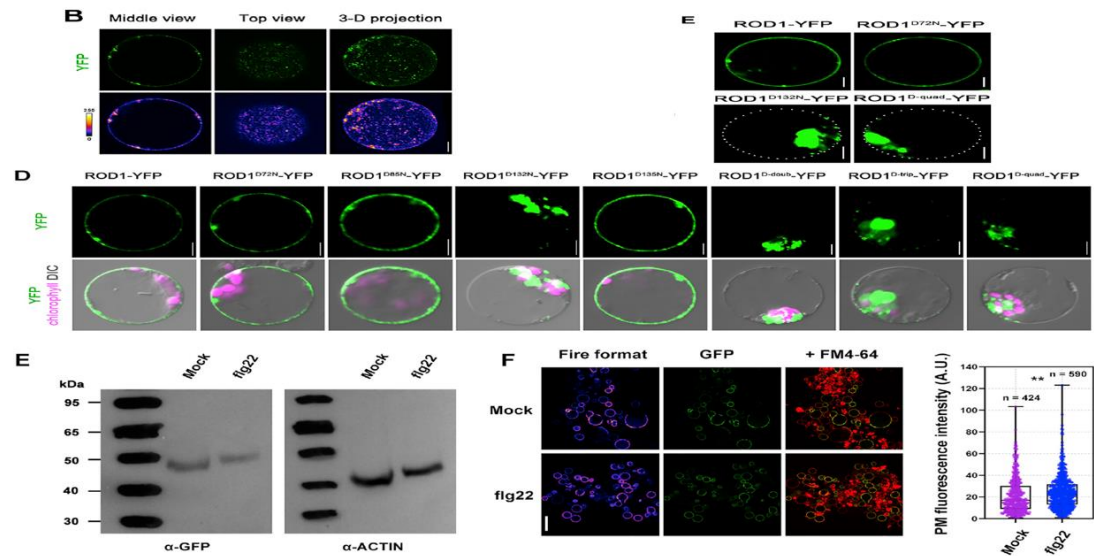


前面提到 **ROD1** 蛋白结构含有 1 个 C2 结构域，那这个有什么用呢？C2 结构域通常是 Ca^{2+} 感受器，这类结构域依赖 Ca^{2+} 与磷脂结合，从而将蛋白质靶向到特定的膜区域。那 **ROD1** 这个蛋白到底能不能和 Ca^{2+} 结合呢？能不能结合那肯定是要有个参照，怎么获得有效的参照呢？对这个蛋白的 4 个氨基酸残基进（因为 C2 结构域的氨基酸残基影响其和 Ca^{2+} 的结合）行突变，有单突变、双重突变、三重突变及四重突变（下图 C）。利用微量热泳动方法来探究 **ROD1** 这个蛋白和 Ca^{2+} 的结合特性（下图 D），图中的 KD 值的大小表示的是分子间的亲和性，越亲和的，其 KD 值越小。没有突变的 **ROD1** 和 Ca^{2+} 的 KD 值很小，而单位点突变和四重突变的 KD 值都很大，整整大了 3 个数量级。 Ca^{2+} 能够改变 C2 结构域的电位，使得 C2 结构与能够与磷脂进行结合。那下一步就是探究 **ROD1** 与不同脂质结合的能力与特异性（下图 A）。从图中可以看到的是，**ROD** 和 **PI(3,5)P2**、**PI(4)P** 及 **PI(5)P** 结

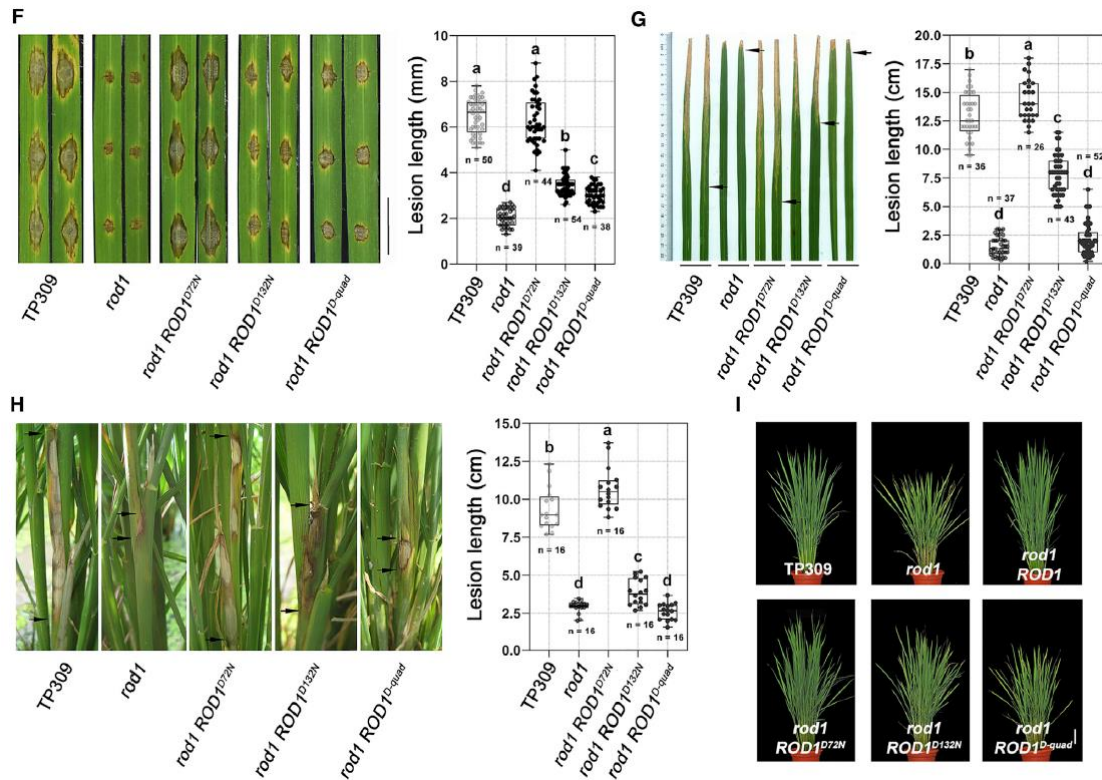
合较好，但是 Ca^{2+} 螯合剂 EGTA 存在的时候，ROD1 就不能与这些脂质进行结合。也就是说当没有 Ca^{2+} 的时候，ROD1 是不能与磷脂进行结合的。



ROD1 这个蛋白在哪里发挥功能呢？这也就需要对其进行定位。通过定位发现这个蛋白位于质膜周围的细胞外围空间（下图 B）。但是将其 4 个氨基酸残基进行突变后都不同程度地出现了蛋白的降解（下图 DE_上）。用 flg22 处理后，ROD1 在质膜上的积累量更多（下图 EF），这说明免疫激活促进了 ROD1 在质膜上的分布。

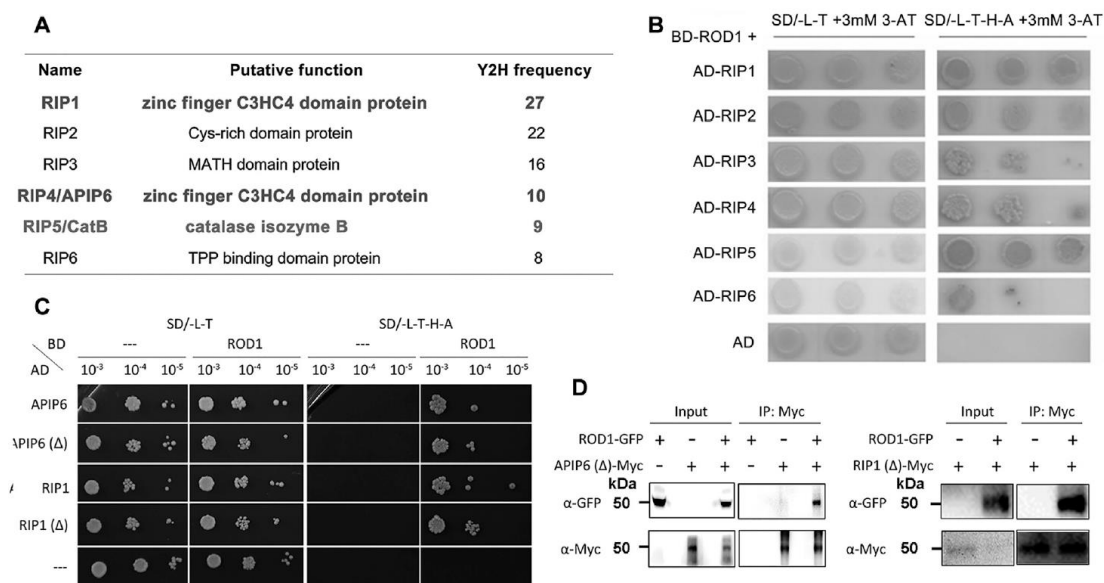


除了利用突变体看突变氨基酸残基后蛋白的定位，还检测突变不同的氨基酸位点后其抗性的变化，也就是过表达这些基因，不同程度的突变都导致对应植株的抗性降低，对三种病害的响应出奇一致（下图 FGH）。不同位点的突变对植株的构型也有一定的影响（下图 I）。

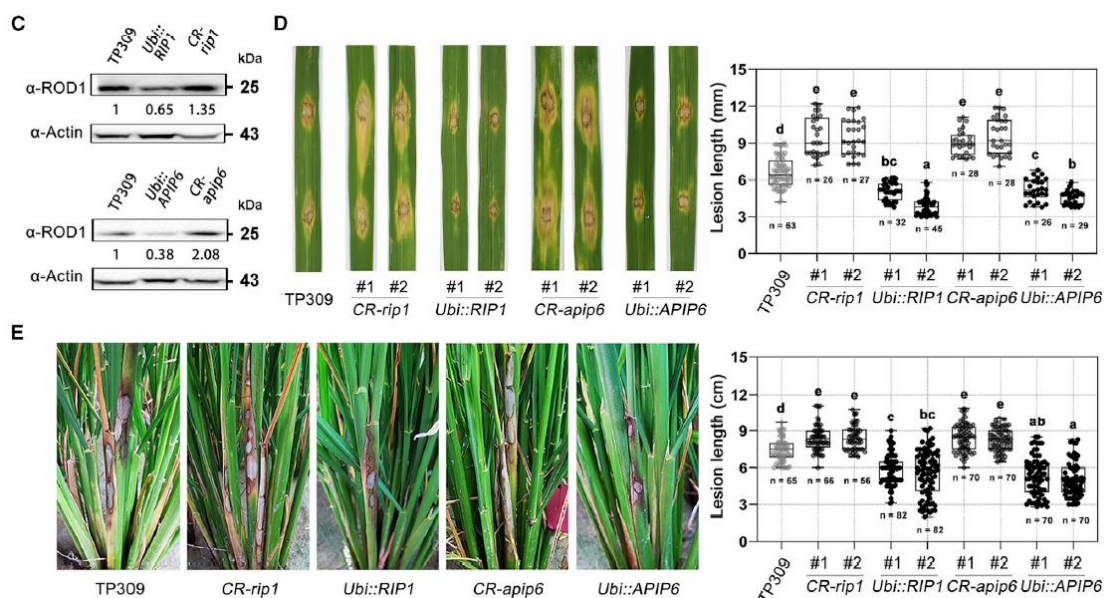


上述这些结果表明 ROD1 和 Ca^{2+} 结合后，进一步与磷脂结合，最后在质膜上“定居”。其实在这个部分已经能够看出，要是没有 Ca^{2+} ，那 ROD1 就“手足无措”了。

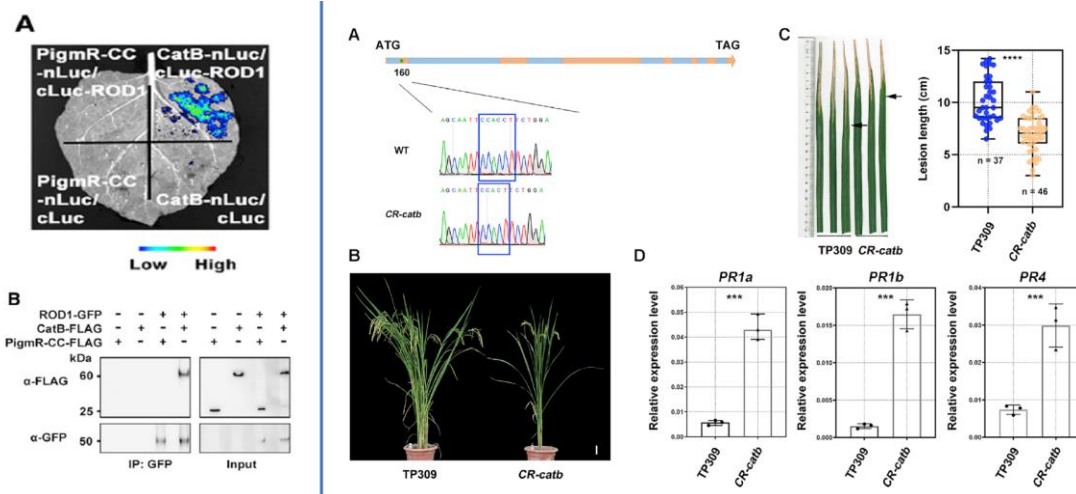
有了材料，有了表型，克隆到了基因，纯化到了蛋白，也知道这个基因能够影响植株的免疫，那下一步就是探究其免疫的机制。利用的方法是酵母双杂（Y2H）。首先是以水稻稻瘟病互作的叶片建立一个 cDNA 文库，然后以 ROD1 为“鱼饵”去钓这个文库中与 ROD1 互作的蛋白。成功得到两个 RING E3 泛素连接酶，分别是 RIP1 和 APIP6（下图）。进行更多试验后发现 ROD1 被 RIP1 和 APIP6 这两个蛋白泛素化（泛素化就是某个蛋白被加上一段标签，被加上标签的蛋白会被其他的细胞器所识别，大多被泛素化的蛋白的最终命运是降解）。



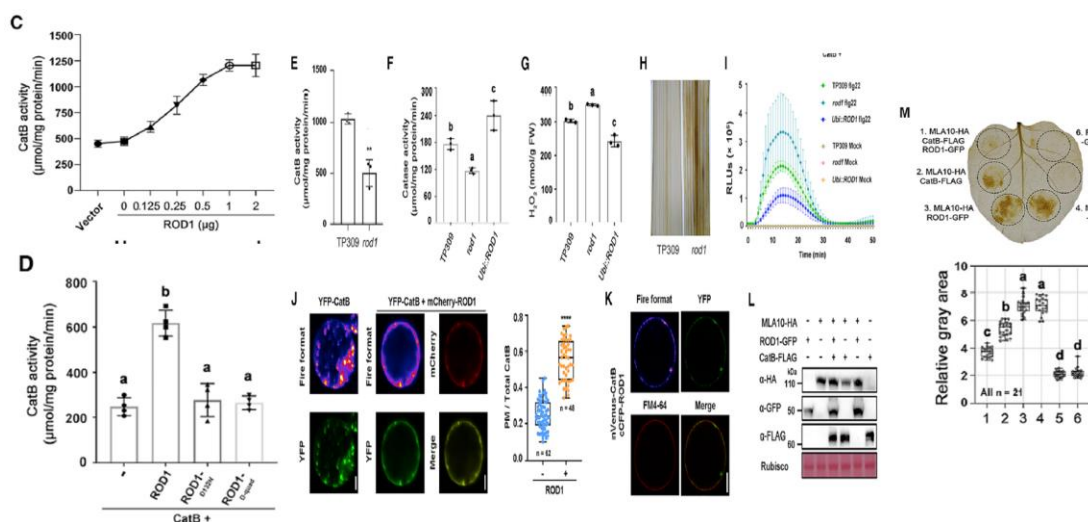
为了进一步验证 RIP1 和 APIP6 的功能，分别构建了敲除系和过表达系。果然不出意外，敲除系的植株的抗性都降低了，而过表达系植株的抗性都提高了。怎么理解呢？将这两个蛋白敲除后，ROD1 就不能被泛素化，那它就不会降解，表达量较高，因此植株的抗性被抑制；将这两个蛋白过表达后，含量增加，ROD1 被“疯狂”泛素化，也就是被疯狂降解，那植株的免疫就不会受到抑制，从而表现出较强的抗性。



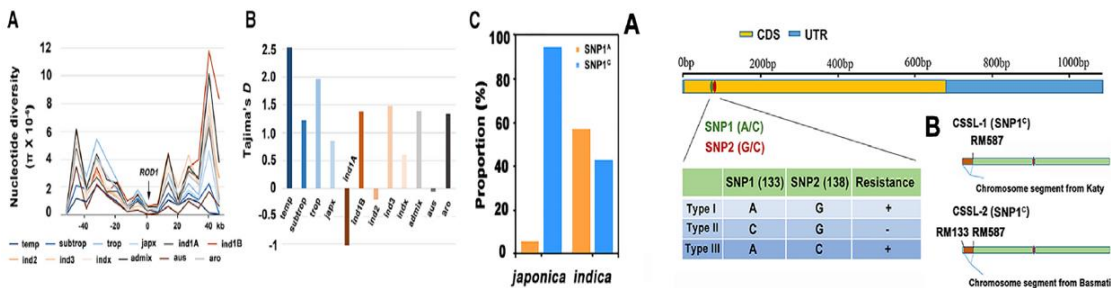
除了 RIP1 和 APIP6 以外，酵母双杂还找到一个和 ROD1 互作的过氧化氢酶 CatB。通过分裂荧光素酶互补试验和 coIP 发现 ROD1 和 CatB 是能够互作的（下图左 AB）。过氧化氢酶能够清除 ROS 从而降低寄主的免疫水平。用 CRISPR/Cas9 将 CatB 敲除后，水稻植株体内 PR 基因的表达上调，植株对病害的抗性增强（下图右 ABCD）。这说明 CatB 负向调控水稻对病害的免疫响应。



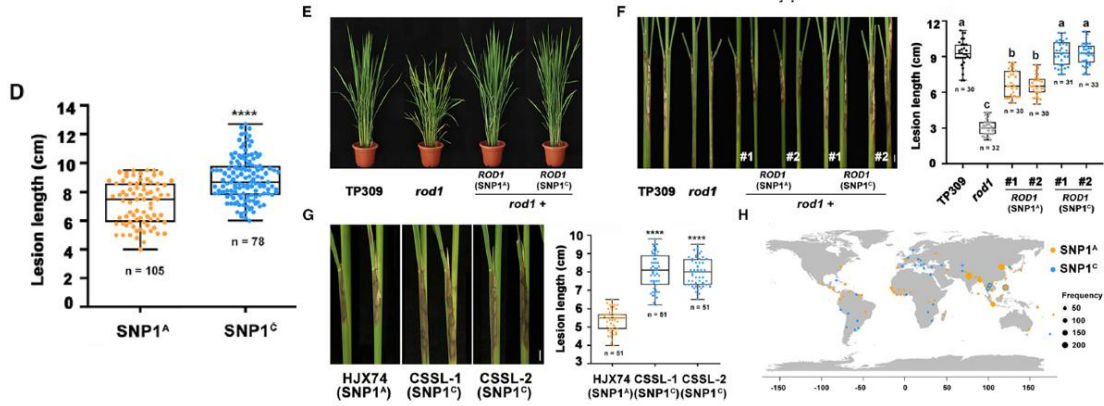
那 ROD1 和 CatB 是如何互作的呢？先用不同浓度的 ROD1 和 CatB 共同清除活性氧，可以发现的随着 ROD1 浓度的提高，CatB 的活性增强，也就是说 ROD1 浓度越高，那 CatB 清除的活性氧也就越多（下图 C）；但是将基因 *ROD1* 进行突变后，蛋白 ROD1 介导的 CatB 清除活性氧的能力没有变化（下图 D），也就是 ROD1 只有和 Ca^{2+} 结合后才能正常和 CatB 互作，从而清除 ROS。怎么理解呢？前面说到，如果突变 *ROD1C1* 结构域上的 4 个氨基酸残基，那 ROD1 和 Ca^{2+} 结合的能力就降低了，没有 Ca^{2+} 结合，ROD1 就不能正常介导 CatB 对 ROS 的清除。进一步提取活体植株体内的 ROD1，发现 TP309 中 CatB 的活性都更搞（下图 EF）。与之对应的是，TP309 植株中 H_2O_2 的含量更低（下图 GHI）。这些结果说明了啥呢？说明 ROD1 是通过介导 ROS 的清除来影响植株的免疫的。CatB 这个蛋白主要定位在过氧化物酶体上，但是呢在质膜上也检测到了 CatB 的存在，当有 ROD1 存在的时候 CatB 在质膜上积累的量更多（下图 JL），这样就说明了 ROD1 促进 CatB 在质膜上的定位。进一步利用本氏烟验证发现 CatB 能够在一定程度上清除由 MLA10 介导的过敏反应，当 ROD1 和 CatB 共同作用的时候这种过敏反应几乎完全消失了。上面这些结果表明 ROD1 首先将 CatB 募集到质膜能赶上，然后再激发 CatB 去清除活性氧。



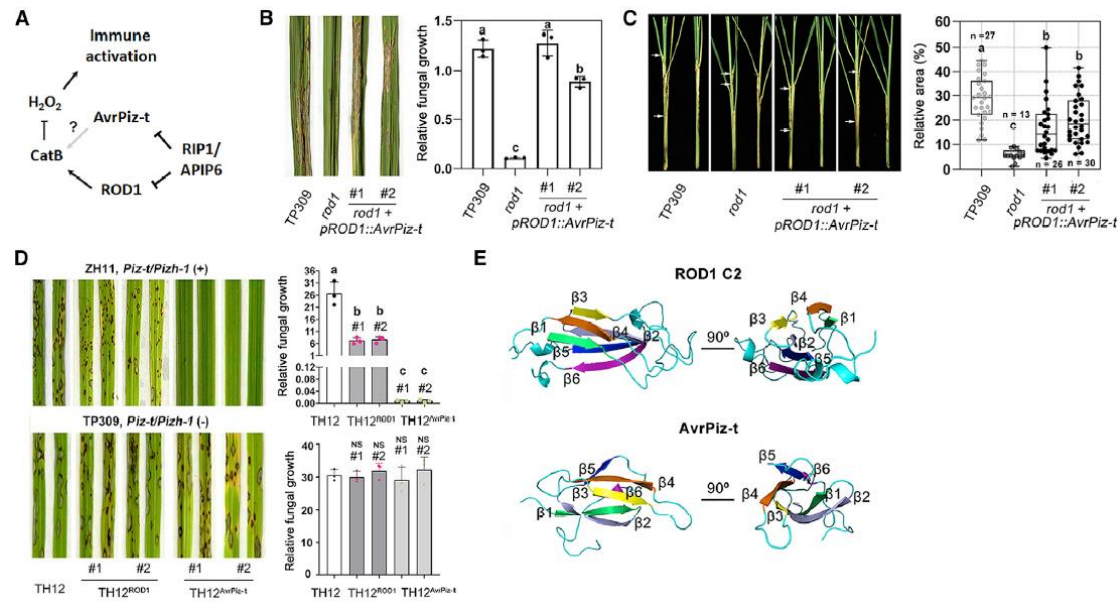
找到这样一个基因，做完上述这些研究，其实已经很完美了。作者他们进一步探究这个基因在水稻群体中的变异情况。通过群体遗传分析发现 *ROD1* 这个位点上的核苷酸多态性很低，也就意味着这个位点在进化过程中发生了选择扫描事件（下图左 A）。在约 3000 个水稻品种中，只有 *indica* 这个亚种的 Tajima's D 值是负值，也就意味着在 *indica* 这个亚种中 *ROD1* 经历了不同的选择作用（下图 B）。在这个基因的编码区发现了两个 SNP，分别是 $SNP1^{A/C}$ 和 $SNP2^{G/C}$ 。第一个 SNP 是一个非同义的突变，在这个位点上脯氨酸变成了苏氨酸；第二个 SNP 是一个同义突变，编码的氨基酸没有发生改变。基于这两种 SNP 分类，可以将其分为三种单倍型（下图右 A）。



随机选了 178 个含有这三种单倍型的水稻品种，调查其在田间对纹枯病的抗性（因为纹枯病还没有主效抗病基因），发现含有 $SNP1^{A/C}$ 的水稻品种对纹枯病的抗性更强（下图 D）。将这两种 SNP 导入到 *rod1* 中后发现 $SNP1^C$ 的抗性基本和 TP309 持平， $SNP1^A$ 植株的抗性比 *rod1* 的弱一些（下图 F），这个结果再次表明 $SNP1^{A/C}$ 确实影响植株的基础免疫。为了进一步验证 $SNP1^A$ 和 $SNP1^C$ 的功能差异，他们构建了染色体片段替换系，发现携带 $SNP1^C$ 的植株的抗性更弱（下图 G），也就意味着携带 $SNP1^A$ 的抗性更强，但是在产量上是没有差异的。进一步探究发现这两种单倍型对 *CatB* 的影响也不一样， $SNP1^C$ 对 *CatB* 的影响更大，也就意味着 *ROD1* 上单核苷酸的变异影响其与 *CatB* 的互作。为了探明 $SNP1^{A/C}$ 的进化差异，比较了 44 个野生稻中 *ROD1* 的序列差异，发现只有 4 个野生稻携带了 $SNP1^C$ ，剩下的 40 个都携带的是 $SNP1^A$ 。 $SNP1^A$ 主要存在于籼稻中，而 $SNP1^C$ 主要存在于野生稻和粳稻中，而籼稻主要分布在热带和亚热带，而粳稻主要分布在温带，也就是说 $SNP1^A$ 和 $SNP1^C$ 的分布是有明显的分布差异的（下图 H）。总之，这些结果表明 *ROD1* 等位基因的变化导致其清除 ROS 的能力有差异，也就导致了品种特异性的抗性差异。



之前鉴定到的与 ROD1 互作的蛋白 APiP6 有报道它能够和稻瘟病菌的效应子 AvrPiz-t 互作并将其降解，RIP1 和 APiP6 高度相似，那 RIP1 能不能与 AvrPiz-t 互作呢？通过一波试验发现这两个蛋白确实能够互作。也就是说 ROD1 与 AvrPiz-t 都与 RIP1 和 APiP6 互作，而且 ROD1 和 AvrPiz-t 都能抑制寄主免疫，那 AvrPiz-t 抑制寄主免疫的机制是不是和 ROD1 一样呢？一波试验过后发现 AvrPiz-t 和 ROD1 一样，与同样的 E3 泛素连接酶互作，参与同样的降解通路，同样促进了过氧化物酶介导的 ROS 清除来抑制寄主植物的免疫。利用 ROD1 的启动子在 *rod1* 中过表达 AvrPiz-t，发现过表达后，*rod1* 原先的抗性基本全部丧失（下图 BC）。然后呢利用两个不同的水稻品种 ZH11 和 TP309 进行进一步验证，其中 ZH11 含有 AvrPiz-t 的抗性基因，而 TP309 没有这个对应的基因，两个水稻品种都接种稻瘟病菌 TH12（没有效应子 AvrPiz-t）。发现 ROD1 能够在一定程度上恢复 TH12 的致病力（下图 D）。再比较 ROD1 和 AvrPiz-t 的蛋白结构后发现这两个蛋白的结构极其相似，也就意味着它们有着一样的免疫抑制机制。



17 基因组学

18 RNA-Seq

19 代谢组学

20 蛋白质组

21 文章发表

22 上课笔记

22.1 李成云：现代植物病理学

22.1.1 20211022

- 韩斌野生稻文章
- 国际水稻所最新的 3K 文章（未发表）
- TIR-NBS-LRR 在水稻中没有，但是 CC-NBS-LRR 在水稻中很多
- DOI: 10.1007/s00122-004-1758-x 水稻上的 R 基因的分布
- [基因资源库](#)
- 水稻白叶枯病菌释放的效应子可以进入到细胞核，与寄主蛋白进行互作
- Pita 的三篇文章可以参考
- Pigm, Pi1、5、9 这些抗性基因只在粳型水稻中存在
- 长期的协同进化过程中，寄主和病原共有的基因！！

23 其他笔记

Gao, Mingjun, Yang He, Xin Yin, Xiangbin Zhong, Bingxiao Yan, Yue Wu, Jin Chen, et al. 2021. "Ca²⁺ Sensor-Mediated ROS Scavenging Suppresses Rice Immunity and Is Exploited by a Fungal Effector." *Cell*.