进化树的可视化脚本分享

——陈军4663

# 脚本介绍

## tree文件绘制svg图

### Itol进行tree可视化转换

* 说明：

效果等同于ITOL官网在线使用的：

<https://itol.embl.de/upload.cgi>

可用于批量生成树图，可以转换成的格式包括png,svg,pdf等

* 来源

官网itolAPI的封装开发:

<https://itol.embl.de/help.cgi#external>

* 集群路径：

/ifs/TJPROJ3/Plant/chenjun/mytools/itol/itol.py

* 使用须知

必须在联网环境下使用。

* 使用方法：

python3 itol.py fipath fopath [--option OPTION] [-h / --help]

**必选参数：**

fipath 输入需要转换的树文件路径, 如 /home/test/test.tre

Fopath 输入转换的树文件路径(无需后缀), 如 out/Mdom 会在当前目录下生成Mdom.svg或其他指定格式

**可选参数：**

**--option "str"**

str部分需用引号引起来

--format 转换格式（默认svg，可多选中间用逗号隔开，可选：svg, eps, pdf, png, newick, nexus, phyloxml）

--display\_mode 显示模式，1 =正常，2 =圆形，3 =无根（1，2，3）

--tree\_x 水平移动像素距离（int）

--tree\_y 垂直移动像素距离（int）

--branchlength\_label\_size 分支长度标签的字体大小，像素为单位（int）

--branchlength\_display 是否显示支长（0, 1）

--bootstrap\_display 是否显示元数据值（0, 1）

...

更多参数请参考官网: <https://itol.embl.de/help.cgi#bExOpt>

* 实例：

**实例1：转换ceshi.tre为myname.svg输出至"out/"下**

/ifs/TJPROJ3/Plant/chenjun/software/python3 /ifs/TJPROJ3/Plant/chenjun/mytools/itol/itol.py /ifs/TJPROJ3/Plant/chenjun/mytools/itol/ceshi.tre out/myname

**实例2：转换ceshi.tre为myname.svg、myname.pdf和myname.png输出至"out/"下，设置参数：选择输出格式，使用圈形，显示支长**

/ifs/TJPROJ3/Plant/chenjun/software/python3 /ifs/TJPROJ3/Plant/chenjun/mytools/itol/itol.py /ifs/TJPROJ3/Plant/chenjun/mytools/itol/ceshi.tre out/myname --option "--format svg,pdf,png,eer --display\_mode 2 --branchlength\_display 1"

### Figtree进行tree可视化转换

* 说明：

可视化tree的另一款软件

* 来源：

官网： <http://tree.bio.ed.ac.uk/software/figtree/>

* 集群路径：

/ifs/TJPROJ3/Plant/chenjun/software/FigTree/FigTree\_v1.4.4/bin/figtree

* 使用须知

软件需要使用本地电脑的图形处理器，如xshell登录时需安装配套使用Xmanager

* 使用方法：

软件自带文档

Usage: figtree [-graphic <PDF|SVG|PNG|JPEG>] [-width <i>] [-height <i>] [-url] [-help] [<tree-file-name>] [<graphic-file-name>]

-graphic produce a graphic with the given format

-width the width of the graphic in pixels

-height the height of the graphic in pixels

-url the input file is a URL

-help option to print this message

Example: figtree test.tree

Example: figtree -graphic PDF test.tree test.pdf

Example: figtree -graphic PNG -width 320 -height 320 test.tree test.png

## 图像处理

### Imgcut白边裁剪

* 说明：

本程序用于图像白边裁剪，优化图片显示。

（若白边过多，后续生成html的显示不好查看，具体表现为每一个框偏大很多）

* 集群路径

/ifs/TJPROJ3/Plant/chenjun/mytools/tools\_tree/imgcut.py

* 使用须知

输入文件或文件夹路径，进行覆盖式裁剪

* 使用方法：

python3 imgcut.py fipath [--filter FILTER] [--bak]

**必选参数:**

fipath 输入需要裁剪的文件路径或文件夹路径

**可选参数:**

-h, --help show this help message and exit

--filter FILTER, -f FILTER

若输入文件(必选参数1)为文件夹路径，则此参数用于文件夹中过滤的后缀名,

可以为任意字符串如“resultbak.png”

--bak, -b 是否使用复制模式，保留原文件，默认为False

* 应用实例：

**# 实例1：裁剪文件test.png**

python3 imgcut.py test.png

**# 实例2：裁剪文件夹中testdir后缀名为.png的文件**

python3 imgcut.py testdir

**# 实例3：裁剪文件夹中testdir后缀名为.bmp的文件**

python3 imgcut.py testdir -f .bmp

**# 实例4：裁剪文件夹中testdir后缀名为.filter.jpg的文件，并保留原文件**

python3 imgcut.py testdir -f .filter.jpg --bak

python3 imgcut.py testdir -f .filter.jpg -b

## Html可视化

### getTree获取标注颜色的树文件

* 说明：

本程序组合了之前itol.py, imgcut.py，在它们基础上开发了上色功能，挑选参考基因标注红色，亲缘关系最近的目标基因标注绿色。

* 集群路径

/ifs/TJPROJ3/Plant/chenjun/mytools/tools\_tree/getTree.py

* 使用须知

参考是按照匹配的基因名字第一个字母来寻找的，设定为非该字母开头的所有基因都为参考基因

* 使用方法：

python3 getTree.py treepath geneName

**必选参数：**

treepath 输入的树文件的路径

geneName 目标基因的第一个字母

* 实例

**# 实例1：将testdir/ceshi.tre于当前文件夹输出标记颜色的svg文件和裁剪的png**

python3 getTree.py testdir/ceshi.tre M

### svg，png在html中可视化

* 说明

本程序用于可视化指定文件夹中的png或svg，生成html

* 集群路径

/ifs/TJPROJ3/Plant/chenjun/mytools/tools\_tree/imgdirView.py

* 使用方法

python3 imgdirView.py dirpath [filter]

**必选参数：**

dirpath 输入的文件夹名

**可选参数：**

filter 筛选文件的后缀，默认为全部文件

* 使用须知

目前仅支持可视化预览png和svg格式的文件，如有需要日后完善。

* 实例

# 实例1：筛选后缀名为t.svg的文件进行可视化

python3 imgdirView.py testdir/ t.svg

# 使用案例

* 说明

该程序遍历该目录下所有的tre文件，按通路名和物种名更名复制于out\_svg，随后对其进行svg生成和上色，最后用html一键展示所有tre结果。

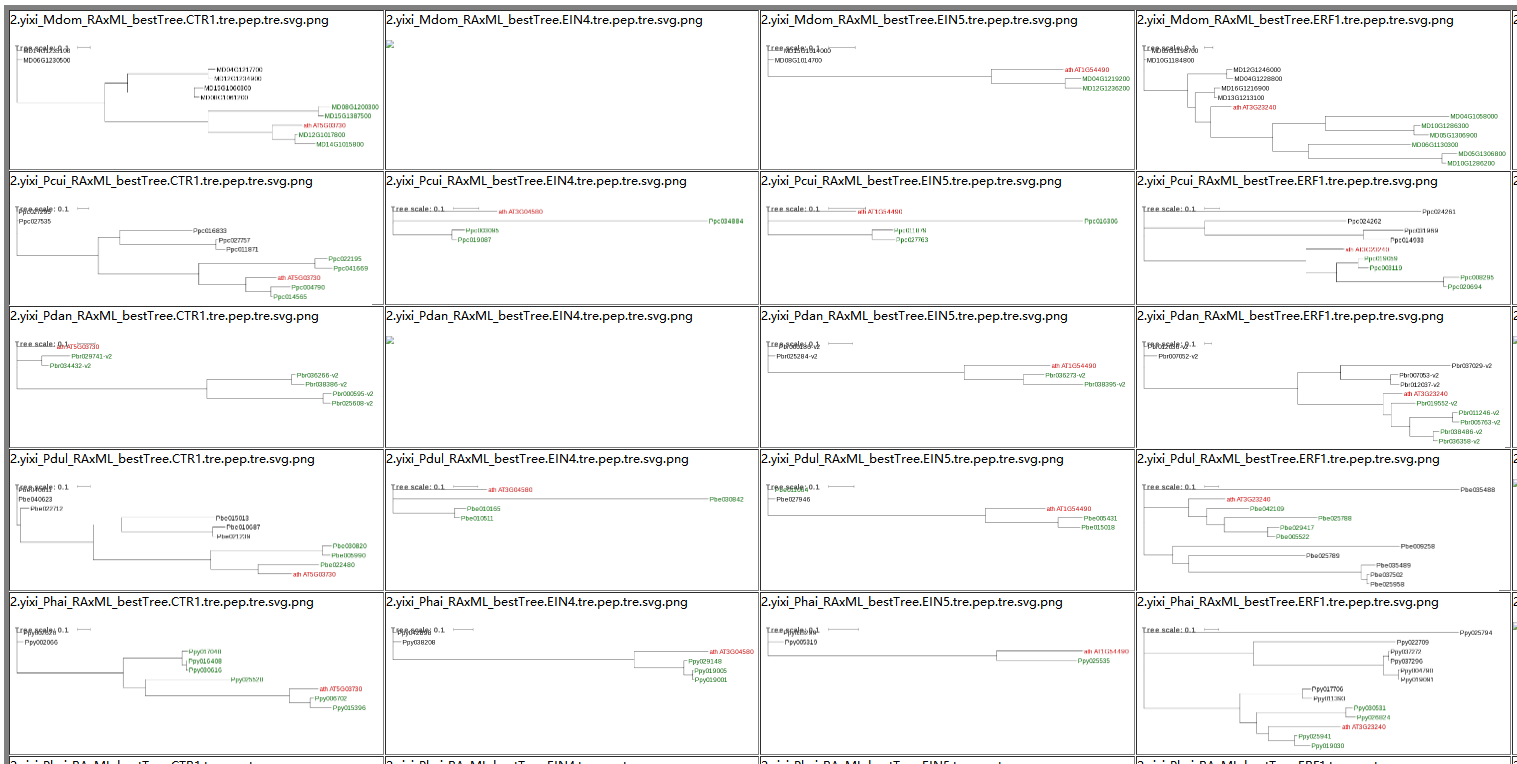
相比之前脚本，主要亮点在于将不同物种可视化为表格。

* 注意事项

脚本基本使用的是非本地计算资源，可以本地直接运行。但注意需要联网

* 可视化效果预览

每一行为不同物种，每一列为相同基因名



* 集群路径

总脚本：

/ifs/TJPROJ3/Plant/chenjun/prj/prj01-nannongli/01.tonglu-guoxing/All\_result/run.sh

需要的输入文件：

1. 包含树的总文件夹，如本项目下的：5.guoxing

2. 从gff提取的每个物种的基本表格信息：list

输出文件：

1. data-table.html 可视化html文件

2. data-table.xls 对应位置是否都有结果

3. out\_svg 输入目录下所有的tre文件集合，及输出结果

5. result.txt 图中标注出的率

4. out\_svg.tar.gz 前面文件的压缩文件，便于下载查看

* 测试路径：

/ifs/TJPROJ3/Plant/chenjun/prj/prj01-nannongli/01.tonglu-new/ALL\_results-3.shanlichun-test

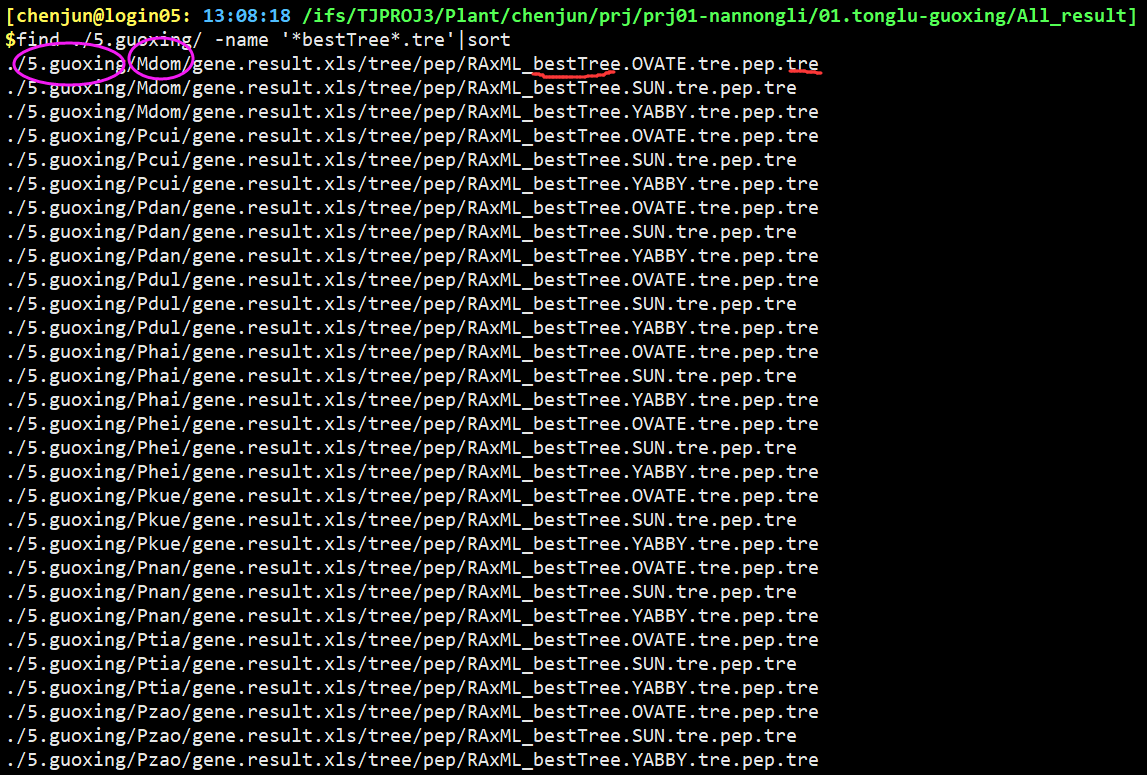
请拷贝该路径下三个文件到自己目录进行运行测试`sh run.sh`

* 步骤详细说明

1. **遍历文件 --> out\_svg**

如图如下目录结构

获取所有文件时，第1和第2个圈分别作为通路名和样本名，复制于当前目录下进行更名，如第一行的文件更名为5.guoxing\_Mdom\_RAxML\_bestTree.OVATE.tre.pep.tre



1. **tre文件转换为svg**

对out\_svg每个文件使用itol.py工具进行输出svg

1. **对svg文件操作进行挑选基因上色**

以物种名第一个字母识别参考基因，对参考基因和挑选目标基因进行红色和绿色着色。

1. **对svg文件转换为png文件**

使用convert命令

1. **对png文件进行白边裁剪**

使用工具包imgcut.py

1. **对out\_svg下的所有png图片嵌入html一个页面可视化**

对每一个物种进行一行显示。

* 评估
* 优

1. 运行速度快，占用资源少
2. 对多个png文件可视化查看方便。

直接下载打开html即可一键查看

* 劣

1. 目录结构受限

输入文件夹必须为上述结构

1. 输入的tre文件基因名受限

物种名第一个字母需与每个tre中挑选的基因第一个字母相同，不同的则判断为参考基因，上色是按照参考基因去寻找最近的