

生物信息学实战



成功不必在我，而功力必不唐捐！

胡适

目录

[版权申明 3](#_Toc453607545)

[Linux 4](#_Toc453607546)

[Linux常用命令 4](#_Toc453607547)

[deb包安装方法 4](#_Toc453607548)

[环境配置 6](#_Toc453607549)

[修改用户环境变量 6](#_Toc453607550)

[安装JDK 6](#_Toc453607551)

[安装Python 6](#_Toc453607552)

[R安装包 6](#_Toc453607553)

[生物信息学软件的使用 7](#_Toc453607554)

[使用MCScanX进行区块共线性分析 7](#_Toc453607555)

# 版权申明

本书仅限于学习交流使用，不得进行其他用途！

# Linux

Linux发行版、Mac OS，讲究的是包管理，模块分离，版本依赖。当你需要安装某个软件，相关依赖的底层模块会被自动安装。Windows讲究的是一个完整的安装包，除了微软自己的.Net Framework、Direct X等通用底层平台外，绝大部分软件都是将自己需要的模块打包在一起安装。

## Linux常用命令

netstat -tupln

列出本机进程监听的端口号

man ascii

显示ascii码表

screen -d -m -S 标识符 命令

后台运行一段不终止的程序，并可以随时查看它的状态。-d -m 参数启动“分离”模式，-S指定了一个session的标识，可以通过-R命令来重新挂载一个标识的session。

python -m SimpleHTTPServer

一句话实现一个HTTP Server，把当前目录设为HTTP服务目录，可以通过http://localhost:8000访问，这也许是这个星球上最简单的HTTP服务器的实现了。

添加用户:

adduser username

为用户添加密码：

passwd username

为用户添加sudo权限

sudo vim /etc/sudoers

添加

username ALL=(ALL:ALL) ALL

## deb包安装方法

deb是debian linus的安装格式，跟red hat的rpm非常相似，最基本的安装命令是：dpkg -i file.deb

dpkg 是Debian Package的简写，是为Debian 专门开发的套件管理系统，方便软件的安装、更新及移除。所有源自Debian的Linux发行版都使用dpkg，例如Ubuntu、Knoppix 等。

以下是一些 Dpkg 的普通用法：

1、dpkg -i <package.deb>

安装一个 Debian 软件包，如你手动下载的文件。

2、dpkg -c <package.deb>

列出 <package.deb> 的内容。

3、dpkg -I <package.deb>

从 <package.deb> 中提取包裹信息。

4、dpkg -r <package>

移除一个已安装的包裹。

5、dpkg -P <package>

完全清除一个已安装的包裹。和 remove 不同的是，remove 只是删掉数据和可执行文件，purge 另外还删除所有的配制文件。

6、dpkg -L <package>

列出 <package> 安装的所有文件清单。同时请看 dpkg -c 来检查一个 .deb 文件的内容。

7、dpkg -s <package>

显示已安装包裹的信息。同时请看 apt-cache 显示 Debian 存档中的包裹信息，以及 dpkg -I 来显示从一个 .deb 文件中提取的包裹信息。

8、dpkg-reconfigure <package>

重新配制一个已经安装的包裹，如果它使用的是 debconf (debconf 为包裹安装提供了一个统一的配制界面)。

# 环境配置

## 修改用户环境变量

$ vim ~/.bashrc

添加如下语句，

export PATH=/home/chenwen/bin:$PATH

保存，退出，更新

$ source ~/.bashrc

## 安装JDK

从Oracle网站下载JDK压缩包，这里我们下载Linux 64位版本。

解压：

$ tar zxvf jdk-8u74-linux-x64.tar.gz

将jdk1.8.0\_74/bin路径添加到用户的环境变量中：

$ vim ~/.bashrc

添加如下一行文本：

export PATH=/your/path/jdk1.8.0\_74/bin:$PATH

保存、退出，更新用户环境变量：

$ source ~/.bashrc

## 安装Python

从Python官网下载Python源码包，这里我们使用Python-3.5.1.tgz。

$ tar zxvf Python-3.5.1.tgz

$ cd Python-3.5.1

$ ./configure --prefix=/home/chenwen/bin/ Python-3.5.1

$ make & make install

## R安装包

install.packages的一个有用的参数dependencies默认是FALSE，如果设置为TRUE的话就会告诉R把当前安装的包所需要的其他包也都下载下来。

# 生物信息学软件的使用

## 使用MCScanX进行区块共线性分析

If you are building on 64-bit you may need to add

Code:

#include <unistd.h>

to msa.h, dissect\_multiple\_alignment.h, and detect\_collinear\_tandem\_arrays.h

MCScanX有三个主程序，在主文件夹中：

MCScanX、MCScanX\_h、duplicate\_gene\_classifier

下游分析程序12个，在downstream\_analyses文件夹中

## 使用goatools进行GO富集分析

### goatools安装

1.安装python开发库

sudo yum install python-devel.x86\_64

2.安装numpy

sudo easy\_install numpy

3.安装fisher

sudo easy\_install fisher

4.安装Graphviz（略过）

5.安装pygraphviz

easy\_install pygraphviz

6.安装goatools

sudo easy\_install goatools

### GO enrichment

find\_enrichment.py --pval=0.05 --indent gene/ehf.list gene.background gene\_go.association

### Map 2 GO slim

map\_to\_slim.py --association\_file=gene\_go.association --slim\_out=direct go-basic.obo goslim\_generic.obo > gene.goslim

## VarScan2

## 使用Trinity进行无参转录组分析

下载

wget –c https://github.com/trinityrnaseq/trinityrnaseq/archive/Trinity-v2.4.0.tar.gz

tar zxvf Trinity-v2.4.0.tar.gz

cd trinityrnaseq-Trinity-v2.4.0

make

运行

## Trimmomatic过滤数据

Trimmomatic使用JAVA运行，支持多线程，数据处理速度非常快，非常适合转录组数据的过滤。由于该软件得到的数据中reads的长度不一，剔除的数据量比较大，不适于基因组的Deno vo数据的过滤。

$ unzip software/Trimmomatic-0.33.zip -d biosoft/

解压缩安装

$ ls biosoft/Trimmomatic-0.33/adapters/

查看安装好的文件

可以看到软件的数据库中已经有了接头序列，其中TruSeq3-PE.fa与TruSeq3-SE.fa为illumina hiseq和miseq的接头，如果不是使用hiseq2000的，到illumina官网下载接头序列并放到这个文件夹中。TruSeq2对应着illumina GAII测序

$ cd train/02.sequencing\_data\_quality\_control/

进入目录

$ mkdir Trimmomatic

新建文件夹

$ cd ~/biosoft/Trimmomatic-0.33/

进入安装的文件夹

$ java -jar trimmomatic-0.33.jar

调试程序

下面对单端测序转录组数据进行过滤

$ java -jar ~/biosoft/Trimmomatic-0.33/trimmomatic-0.33.jar SE -threads 4 -phred33 ~/train/00.incipient\_data/data\_for\_gene\_prediction\_and\_RNA-seq/000\_rep1.fastq 000rep1.fastq ILLUMINACLIP:/home/tempuser/biosoft/Trimmomatic-0.33/adapters/TruSeq3-SE.fa:2:30:10 LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:36

参数：

单端测序数据SE

线程数4

phred33格式

输入 输出

接头序列：允许最大的mismatch数，回文匹配碱基数阈值，简单匹配碱基数阈值

LEADING头部去掉质量低于3的

TRAILING尾部去掉质量低于3的

每4个减基是一个阅读框，假如4个平均质量低于15，就删掉

36是reads最小长度，短于这个就不要

运行好了，会显示过滤了多少

下面对双端测序转录组数据进行过滤

$ java -jar ~/biosoft/Trimmomatic-0.33/trimmomatic-0.33.jar PE -threads 4 -phred33 ~/train/00.incipient\_data/data\_for\_genome\_assembling/fragment.1.fastq ~/train/00.incipient\_data/data\_for\_genome\_assembling/fragment.2.fastq fragment.1.fastq fragment.1.fastq.unpaired fragment.2.fastq fragment.2.fastq.unpaired ILLUMINACLIP:/home/tempuser/biosoft/Trimmomatic-0.33/adapters/TruSeq3-PE.fa:2:30:10 LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:36

双端是PE，其他参数一样，输入为两个序列，输出还多了一个没配对的数据

运行完成

$ ll -h

查看生成的结果，

使用下面命令可以批量对几十上百个转录组进行过滤，反字符//用于删除，i用于循环，变量使用$,真正斜线用反斜线表示：

$ for i in `ls ~/00.incipient\_data/data\_for\_gene\_prediction\_and\_RNA-seq/\*.fastq`

> do

> x=${i/\*\//}

> echo java -jar /opt/biosoft/Trimmomatic-0.33/trimmomatic-0.33.jar SE -threads 4 -phred33 $i $x ILLUMINACLIP:/opt/biosoft/Trimmomatic-0.33/adapters/TruSeq3-SE.fa:2:30:10 LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:36

> done > command.trimmomatic.list

$ sh command.trimmomatic.list

批量运行

也可以这样运行

$ chmod 755 command.trimmomatic.list

$ ./command.trimmomatic.list