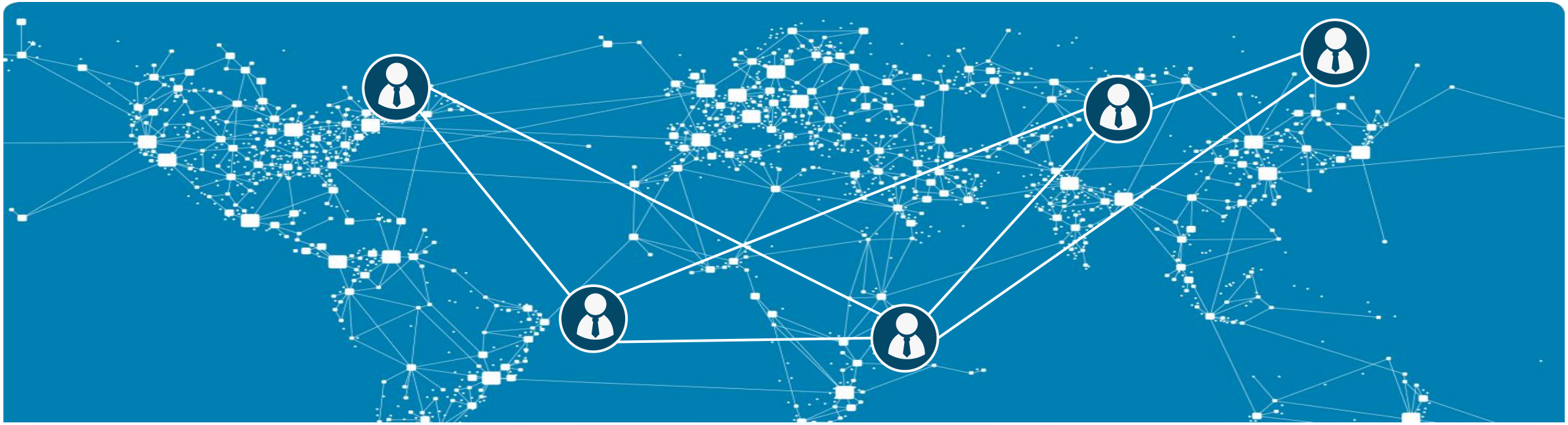
2020微生物组—扩增子16S分析研讨会第10期



12Linux简介和实操

易生信

2020年12月4日

学习方式

- 提前预习
- 仔细听讲
- 先运行再理解
- 紧跟步伐，跟不上的及时在课堂提出或寻找助教老师解决
- 课后复习，基础知识学习靠背和反复练
- 书读百变，其义自见
- 码敲十遍，不会也难

目录

- Linux系统简介
- Linux运行环境
- Linux常用命令

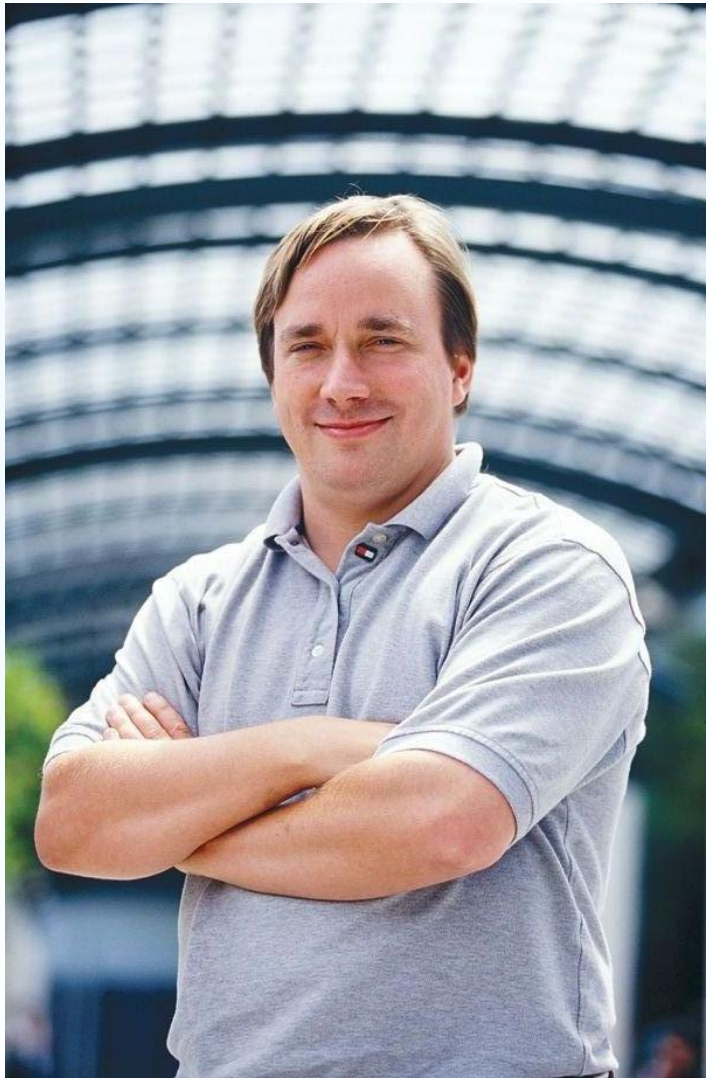
http://www.ehbio.com/Training/index.php/Home/Index/Articles.html

Linux是什么?

- Linux是什么？
  - Linux是一种操作系统
  - 多用户、多任务
  - 与UNIX类似 (MacOS是基于UNIX)
- Linux在哪儿？
  - 网站、数据库
  - 计算服务器
  - 台式机/笔记本
  - 路由器
  - 智能手机Android

Linux简史

- 1991年10月5日
  - Linus Torvalds(林纳斯·托瓦兹)首次发布了Linux 0.02，Linux诞生
  - 因为学校的UNIX主机总是排队
  - Linus就想在自己的电脑上"山寨"一个UNIX，这样就不用排队了
- 从Minix受到启发
  - 谭宁邦教授为了进行UNIX操作系统教学，编写了一个微型的用于示范的UNIX系统，称为Mini UNIX，简称Minix
  - Minix性能优异，但是过于简单，且教授没有时间维护
  - Linus受到了Minix的源代码启发，在其基础上进行改进，完成了Linux，并在他自己的386个人计算机上成功运行
- Linux的发展
  - 不断加入新功能：以完整兼容UNIX和不同的硬件平台
  - 网络虚拟团队的产生：代码开源化，维护社区化
  - 产生定期发布内核的网站：kernel.org，并规范了版本号
  - 众多公司认识到Linux的商业价值：纷纷参与开发Linux



为什么学习Linux?

- 系统开源免费——节约成本且更安全
- 90%以上服务器为Linux系统
- 长期运行的稳定性
- 多数生物学软件只有Linux版本
- 强大的Bash命令简化繁琐的操作，尤其是大大简化重复性工作

Linux命令行运行环境

MINGW64:/c/Users/woodc/Desktop

```
woodc@DESKTOP-ONKLVFL MINGW64 ~/Desktop
$
```

GitForWindows

yongxin@yongxin: ~

```
Welcome to Ubuntu 20.04 LTS (GNU/Linux 4.4.0-18362-Microsoft x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advanta

  System information as of Wed Aug 26 23:07:4

  System load:    0.52      Processes:
  Usage of /home: unknown   Users logged in:
  Memory usage:   71%       IPv4 address for
  Swap usage:     0%        IPv4 address for

0 updates can be installed immediately.
0 of these updates are security updates.


The list of available updates is more than a
To check for new updates run: sudo apt update


This message is shown once once a day. To dis
/home/yongxin/.hushlogin file.
(base) yongxin@yongxin:~$
```

Ubuntu

XShell

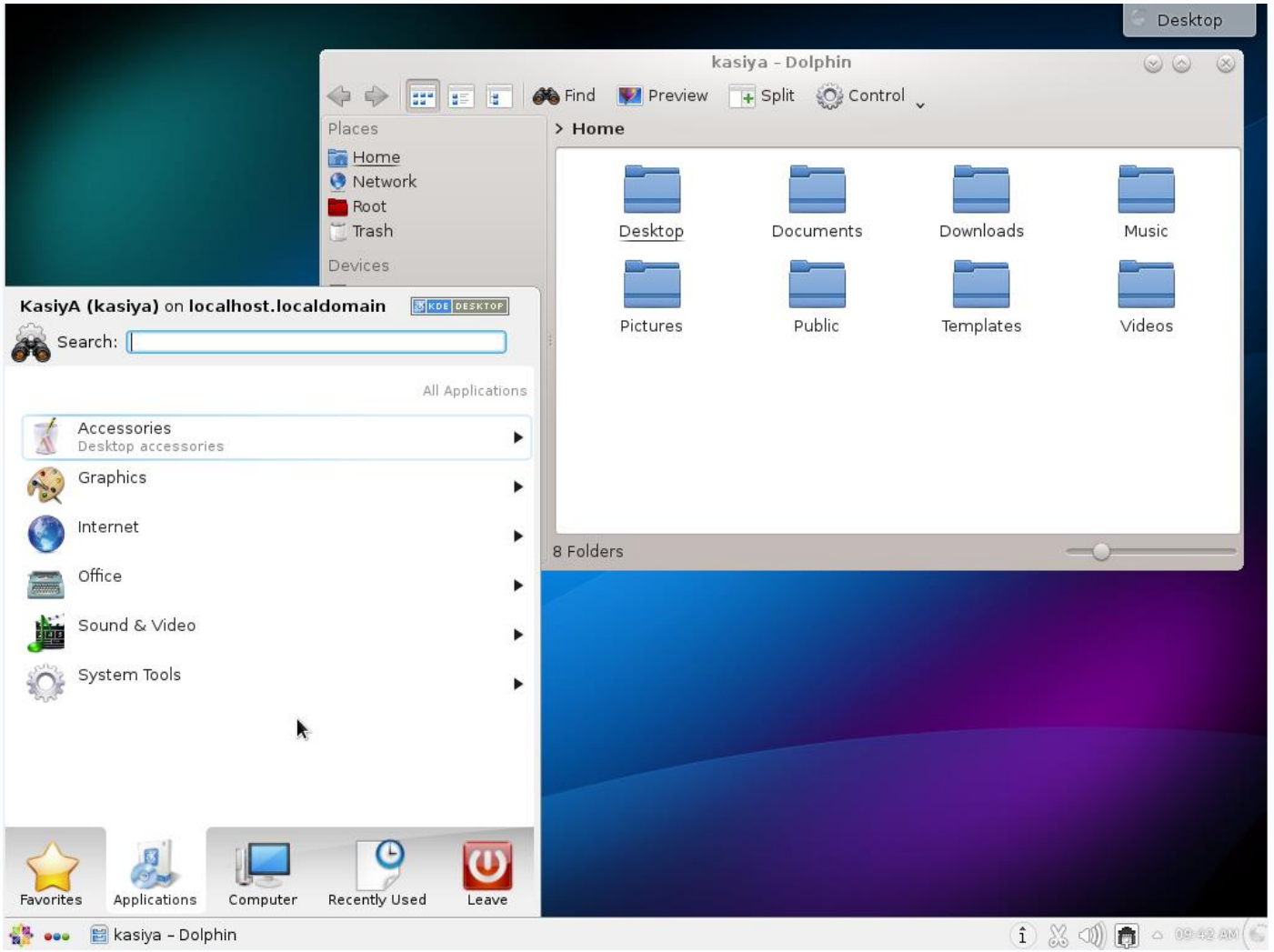
bailab-pub - Xshell 5 (Free for Home/School)

1 bailab-pub

```
Tasks: 979 total,   2 running, 977 sleeping,   0 stopped,   0 zombie
%Cpu(s):  1.9 us,  0.1 sy,  0.0 ni, 98.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem : 10567488+total, 14869265+free, 13685348 used, 89437088+buff/cache
KiB Swap: 10736353+total, 10736133+free,    21984 used. 10402117+avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
 8291 xiaoning  20   0 12.675g 9.844g  32248 R  94.4  1.0   1804:35 python3
71409 public    20   0   45140   4412   3060 R  11.1  0.0   0:00.04 top
 3589 root      20   0 2216796  65660  32428 S   5.6  0.0  64:23.98 dockerd
    1 root      20   0  120156   6260   4084 S   0.0  0.0   1:51.97 systemd
    2 root      20   0       0      0      0 S   0.0  0.0   0:00.49 kthreadd
    3 root      20   0       0      0      0 S   0.0  0.0   0:46.87 ksoftirqd/0
    5 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 kworker/0:0H
    8 root      20   0       0      0      0 S   0.0  0.0  37:11.01 rcu_sched
    9 root      20   0       0      0      0 S   0.0  0.0   0:00.00 rcu_bh
   10 root      rt   0       0      0      0 S   0.0  0.0   0:02.62 migration/0
   11 root      rt   0       0      0      0 S   0.0  0.0   0:05.32 watchdog/0
   12 root      rt   0       0      0      0 S   0.0  0.0   0:05.23 watchdog/1
   13 root      rt   0       0      0      0 S   0.0  0.0   0:02.70 migration/1
   14 root      20   0       0      0      0 S   0.0  0.0   0:06.04 ksoftirqd/1
   16 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 kworker/1:0H
   18 root      rt   0       0      0      0 S   0.0  0.0   0:05.11 watchdog/2
   19 root      rt   0       0      0      0 S   0.0  0.0   0:02.73 migration/2
[public@biocloud:~]$
```

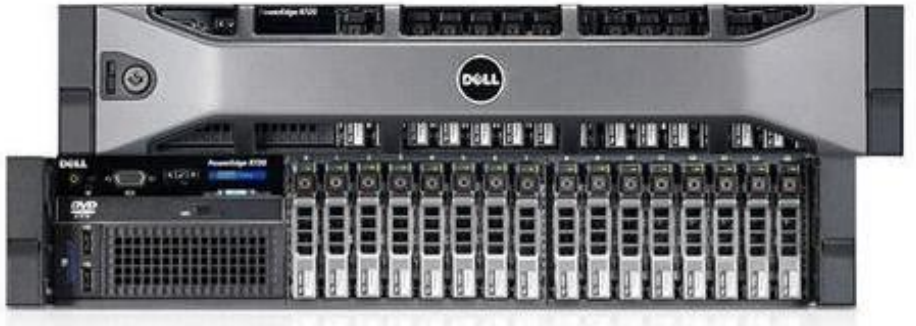
Linux界面 – 图形用户界面



计算平台的选择

- 工作站：相当于日常使用的高配置台式机和笔记本，价格5 K – 30 K，可实现通常 1 – 10 GB规模数据的分析。

- 服务器：专业的主板，可以安装CPU X 1~4 + 内存 X 48 + 磁盘阵列 (8T x 12)，价格30 K – 1 M，处理 10 GB – 10 TB规模数据。



- 计算集群：服务器 X N。集群中成员根据配置特点分为管理结点、登陆结点、计算结点和胖结点，扩增子分析样本不多可使用计算结点，宏基因组拼接内存占用多，常使用胖结点。

分析所需硬件配置

- 扩增子分析工作站：预算10 - 50 K起

CPU: Intel i7 / AMD锐龙 6 ~ 32核　　内存：32 ~ 128 GB

磁盘：1~3 x 8 TB　　OS：Ubuntu 18.04

- 宏基因组分析服务器：预算 30~700 K

CPU: Intel(R) Xeon(R) 10 ~ 28核 X 4颗 (最多112核，224线程)

内存：512GB ~ 2T(1 TB = 1024 GB)

磁盘阵列：48 ~ 96 TB X 2

OS：Ubuntu 18.04 / CentOS

目录

Linux运行环境

- Linux服务器或云，采用ssh终端或网页Rstudio远程登陆
  专业高效的工作方法
- Windows安装gitforwindows实现部分Linux命令
  Windows中上实现上百个Linux命令，无法安装Linux软件，简单数据分析
  - Windows轻松实现linux shell环境：gitforwindows
- Win10内置Linux子程序
  指定版本可安装、可安装软件、效率高
  - Windows10安装Linux子系统Ubuntu 20.04LTS，轻松使用生信软件，效率秒杀虚拟机
- 在Windows中安装Virtualbox实现
  体积大、运行效率低，较真实的Linux环境
  适合学习和小规模数据处理
  QIIME虚拟机安装配置及挂载外部目录

Linux界面 – 命令界面

- 在图形界面下打开 Terminal或远程登录服务器会看到类似如下的界面，这是我们大部分时候的工作环境。

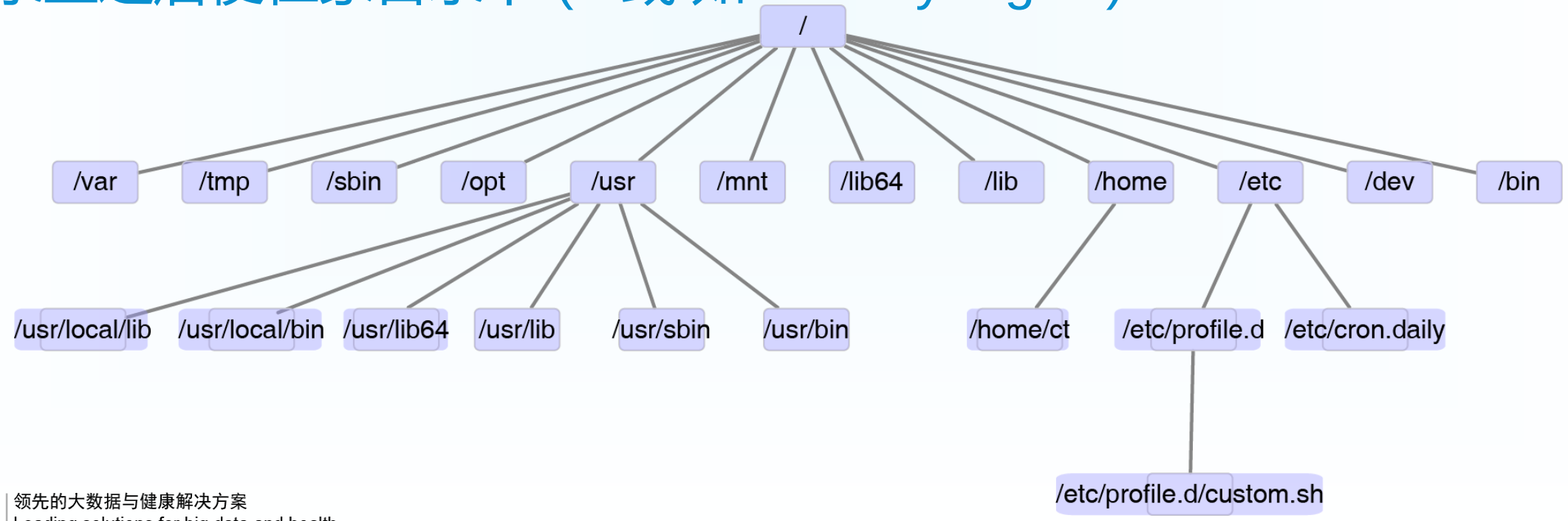
```
Last login: Mon Jun  5 16:56:56 2017 from 239.241.208.209

Welcome to aliyun Elastic Compute Service!

ct@ehbio:~$
```

Linux目录结构

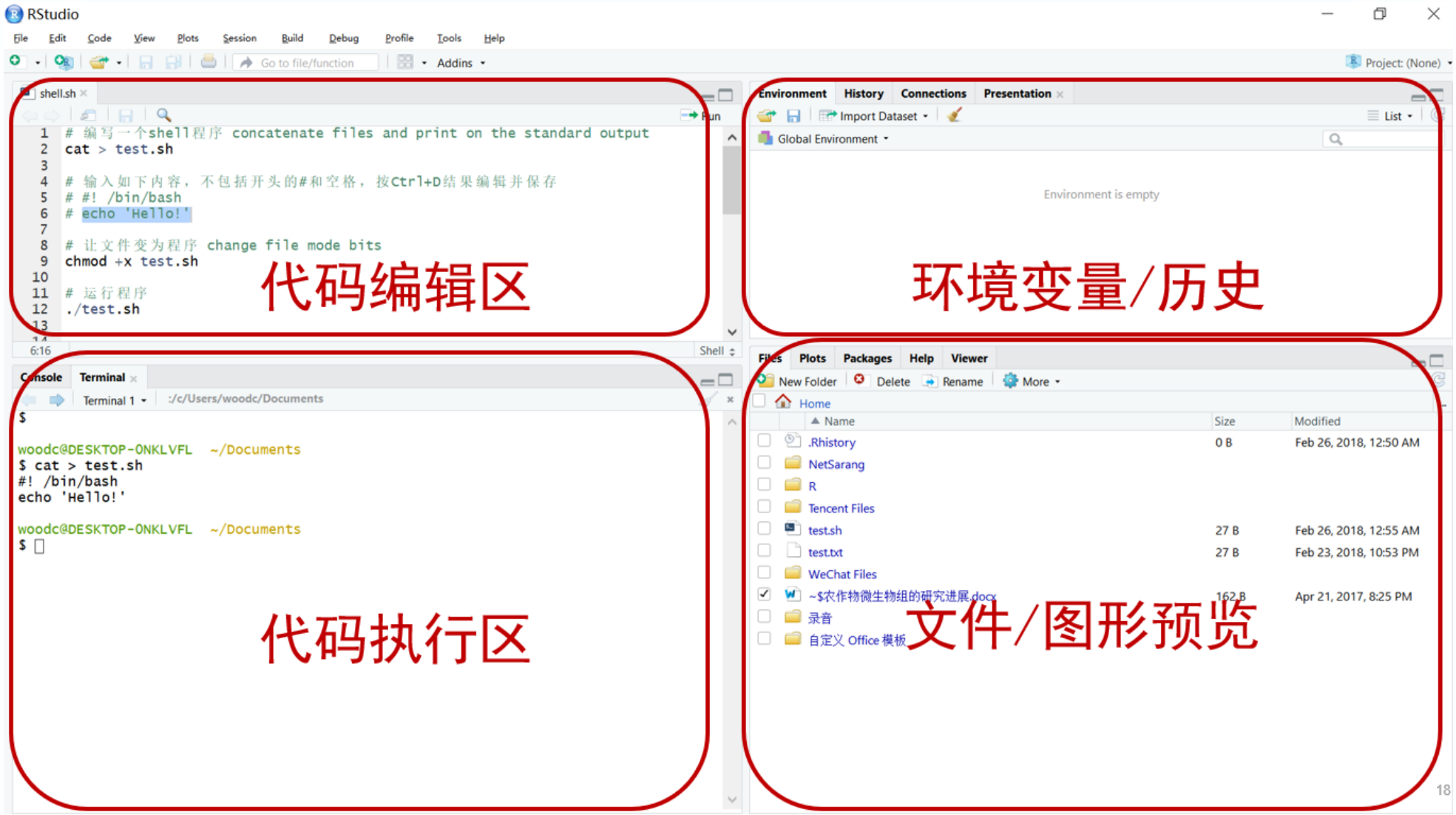
- Linux下所有目录都在根目录下，用 / 表示。下面有几个固定的子文件夹，/bin, /tmp, /usr, /home, /etc等。因此在访问目录时一般加上 / 指示相对于绝对路径。
- 登录上之后便在家目录下 (~ 或 如/home/yongxin)



目录结构

- **/**：根目录，所有的目录、文件、设备都在/之下，/就是Linux文件系统的组织者，也是最上级的领导者。
- **/bin**：bin 就是二进制（binary）英文缩写。在一般的系统当中，都可以在这个目录下找到linux常用的命令。系统所需要的那些命令位于此目录或/usr/bin、/usr/local/bin
- **/etc**：etc (Editable Text Configuration)目录是linux系统中最重要的目录之一。目录下存放了系统管理时要用到的各种配置文件和子目录。如要用到的网络配置文件，文件系统，系统配置文件，设备配置信息，设置用户信息等都在这个目录下。
- **/home**：如果建立一个用户，用户名是"xx",默认在/home目录下就有一个对应的/home/xx路径，用来存放用户的主目录。当然也可指定新用户家目录在其它位置。
- **/mnt**：这个目录一般是用于存放挂载储存设备的挂载目录的，比如磁盘阵列、U盘和移动硬盘、Windows下的c/d盘等目录。
- **/lib**：lib是库（library）英文缩写。这个目录是用来存放系统动态链接共享库的。几乎所有的应用程序都会用到这个目录下的共享库。因此，千万不要轻易对这个目录进行什么操作，一旦发生问题，系统就不能工作了。

Win10 + Rstudio + gitforwindows运行Linux命令



目录

Linux基本命令

- 开始执行命令
  - 在Linux下，我们都是通过"命令"进行操作的
  - 命令 = 执行某一个或某一组程序
  - 执行命令，需要在"命令行"输入输令，并按回车执行。

  - [ngs0@localhost ~]$ 命令 [-选项] 参数1 参数2 ……

  - 说明：
    - 命令行永远以可执行程序开始
    - [-选项] 的方括号表示该项目是可选的，不是每次都必须要输入
    - 不同的项之间以空格分隔，命令行以回车结束并即刻执行
    - Linux是区分大小写的，即cd和CD是不同的意义

熟悉工作环境

- pwd # 显示当前工作目录 (print working directory)

pwd为命令，井号(#)后内容为注释内容，方便读者理解

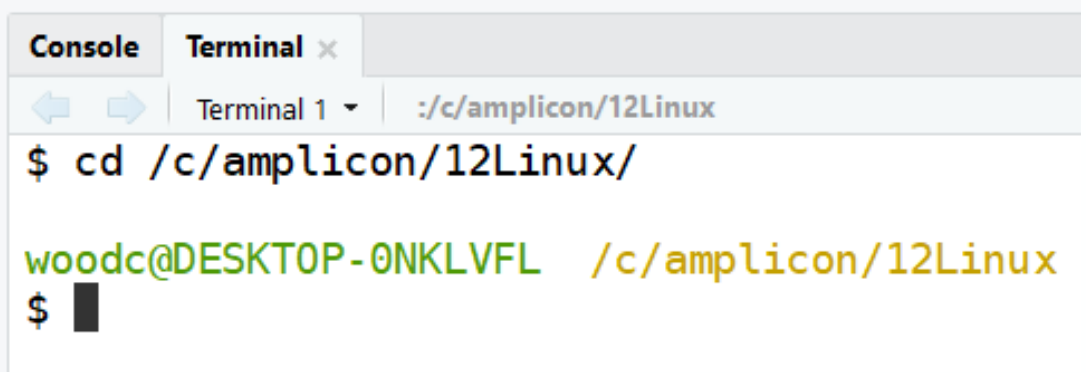
pwd

```
woodc@DESKTOP-0NKLVFL  ~/Documents
$ pwd
/c/Users/woodc/Documents
```

- 我们将易生信扩增子U盘中amplicon目录复制到C盘根目录

cd 切换本节工作目录 (change dir)

cd /c/amplicon/12Linux/



常用命令-文件操作

- ls # 显示当前文件夹文件 (list)

ls -l # 列表显示

ls -l dir # 显示目录dir下的文件

- mkdir # 新建文件夹 (make directory)

mkdir test # 创建目录

- cd # 切换目录 (change dir)

cd test #进入test目录

cd .. # 后退到上一级目录，记住 . 代表当前目录，.. 代表上一级目录

cd ./test 等同于 cd test，一般省略 ./

```
woodc@DESKTOP-971FRV6  /c/amplicon
$ ls
.RData                                  24Diversity.zip
.Rhistory                               25QIIME2/
11Platform/                             26Network/
12Linux/                                31PICRUSt/
13R/                                    32FAPROTAX/
14Figure/                               33MachineLearning/
16AI/                                   34Evolution/
21扩增子分析背景知识介绍.pdf  35ENVFactor/
22Pipeline/                             36AI/
23LEfSe/                                41Question/
23STAMP/                                Metagenome/
24Compare/                              PPT/
24Diversity/                            Video/

woodc@DESKTOP-971FRV6  /c/amplicon
```

复制、移动/改名，删除

- cp # 拷贝文件，原文件至目标位置 (copy)

cp test.sh file_temp.txt # 复制文件

cp test.sh test/ # 复制文件到指定目录

- mv # 移动或改名文件 (move)

mv test.sh temp.sh # 移动，不更改目录则为改名

- rm # 删除文件 (remove)

rm test/test.sh # 文件

rm -r test # 删除文件夹

快捷键：Tab键自动补全或提示相关结果

- Tab键：
  - 可用于补全：命令、文件名，因此只需记住一部分名称即可
  - 补全的原则：
    - 如果唯一，则按一次Tab直接补全
    - 多个选择时，按一下补至最大唯一，再按两次Tab显示可能提示

如目录存在t开头唯一文件test.sh

cat t 并按tab键

即可显示如下：

cat test.sh

多个选择时，按一下补至最大唯一，再按两次Tab显示可能提示

```
$ ls  1

woodc@DESKTOP-0NKLVFL  /c/amplicon/12linux
$ ls 12linux简介与实操.p

woodc@DESKTOP-0NKLVFL  /c/amplicon/12linux
$ ls 12linux简介与实操.p
12linux简介与实操.pdf   12linux简介与实操.pptx

woodc@DESKTOP-0NKLVFL  /c/amplicon/12linux
$ ls 12linux简介与实操.p
```

快捷键：中止命令Ctrl+C

- Ctrl-C
- 功能：用于终止当前运行中的命令
- 用法：在程序运行中，先按住Ctrl键不放，然后再按C键，释放
- 例如：执行ping -t www.baidu.com↙
- 按Ctrl-C终止ping的循环，退回到Linux命令提示符

```
$ ping -t www.baidu.com

正在 Ping www.a.shifen.com [119.75.216.20] 具有 32 字节的数据：
来自 119.75.216.20 的回复：字节=32 时间=7ms TTL=55
来自 119.75.216.20 的回复：字节=32 时间=5ms TTL=55
来自 119.75.216.20 的回复：字节=32 时间=7ms TTL=55
来自 119.75.216.20 的回复：字节=32 时间=5ms TTL=55

119.75.216.20 的 Ping 统计信息：
    数据包：已发送 = 4，已接收 = 4，丢失 = 0 (0% 丢失)，
往返行程的估计时间(以毫秒为单位)：
    最短 = 5ms，最长 = 7ms，平均 = 6ms
Control-C
```

编写一个Shell脚本/程序

- cat # 查看文件 (concatenate files and print)

cat > test.sh # 创建文件，并开始写入，按Ctrl+D结束输入

```
# ! /bin/bash
echo "Hello YSX !"
```

- chmod +x # 添加文件可执行权限 rwx读、写、执行 (change file mode bits)

chmod +x test.sh

- 运行程序，执行结果为Hello!

./test.sh

```
$ ./test.sh
Hello!
```

认识二代测序文件格式-Fastq

Illumina测序数据通常是Fastq格式。如果是双端测序，会有两个文件

*_1.fq.gz　*_2.fq.gz

```
@EAS139:136:FC706VJ:2:2104:15343:197393 1:Y:18:ATCACG
GCTCTTTGCCCTTCTCGTCGAAAATTGTCTCCTCATTCGAAACTTCTCTGT
+
IICFFFDEHHHHFIJJJ@FHGIIIEHIIJBHHHIJJEGIIJJIGHIGHCCF
```

每4行表示一条reads

第一行：@序列ID，包含index序列及read1或read2标志：

第二行：碱基序列，大写"ACGTN"；

第三行："+"，预留行，有时候是序列ID；

第四行：质量值序列：字符的ASCII码值-33=质量值

FASTQ格式解释和质量评估 2fasta&fastq

常用序列存储格式fasta

基本概念——相关格式：

- FASTA：文件中以">"起始一条序列/read

```
>b1.1_0
TACGGAGGATCCGAGCGTTATCCGGATTTATTGGGTTTAAAGGGAGCGTAGG
GCGGCTCAACCGTAAAATTGCAGTTGATACTGGGTGTCTTGAGTACAGTAGA
AAATGCTTAGATATCACGAAGAACTCCGATTGCGAAGGCAGCTTGCTGGACT
GGTATCAATCAGG
```

- FASTQ：文件中每4行代表一条read，"行开头@"

```
@ST-E00142:187:HJFGGCCXX:1:1101:7425:1590 1:N:0:TTCACG
NAGGATGAAGTCTGGGTAACACCAGATGGAGGTCCGCACCAATAAGCGTTGAAAAGCT
+
#AA,<A<FA<FFAFKKKKKKF,FFKFKKKFFKKFFAAAF,<FFAAF,,,,77F7AFA,
```


fasta&fastq格式详解

Fastq文件操作

- 我们提供了一个fastq文件，example.fq.gz
- gz结尾的格式为压缩文件，使用gunzip命令解压文件 (g-un-zip)

gunzip example.fq.gz

- head/tail显示文件头尾

head example.fq

tail example.fq

```
$ head example.fq
@KO1.1
GTAGTCCACGCCGTAAACGATGGATGCTAGCCGTTGGCCGGTTTACCGGTCAGTGGCGCAGCTAACGCTTTAAGCATCCCGCCTGGGGAGT
ACGGTCGCAAGATTAAAACTCAAAGGAATTGACGGGGGCCCGCACAAGCGGTGGAGCATGTGGTTCAATTCGACGCAACGCGAAGAACCTT
ACCAGCTCTTGACATGTCTCGTATGGGTTTCAGAGATGAGACCCTTCAGTTCGGCTGGCGAGAACACAGGTGCTGCATGGCTGTCGTCAGC
TCGTGTCGTGAGATGTTGGGTTAAGTCCCGCAACGAGCGCAACCCTCGCCTTTAGTTGCCATCATTTAGTTGGGCACTCTAAAGGGACTGC
CGGTGATAAGCCGCGA
+
IIIIIHIIIIIIIIGIGIHHIHHIIIIIIIIIIIIIIIIIIIHIIIIIHHIIIHHGIDHDHHIIIIIIIIIIIHIIIGHIHIHHHHIHHII
IHHDGHIHHHDCHHIIIGIIIIIHIGHIIHHHGHEHDHHIGIIIIGIHIHHHI.BEHJJJJJ?JJJJJJJJJJJJJJJJJJJJJFJJJJ
JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJGHGB-CHAG.AA-EHHHHHCIIHHEHHFHIIIHEHFIIGIIIHGHIGIIHEIIII
HHHHHIGIIIHIIIHHGHCHFHHDHHIHIHHHHHCIGIHHHHCHIHIIIHHGIIIIIIIIHGHHFHGHIIHGIIHEHIIIHHHIIIIHIHH
IHIIIIIIIIHIGIIH
```

- 按页查看文件, -S不换行，空格翻页，q退出

less -S example.fq

转换fastq为fasta，并按顺序重命名序列

```
# 转换fastq为fasta，并按顺序重复命名序列
awk 'NR%4==2 {print ">E"NR/4+0.5"\n"$0}' example.fq > example.fa
# 显示fasta文件末尾10行
tail example.fa
```

```
>E2498
GTAGTCCACGCCATAAACGATGAGGACTAGACGTTGGAGGGGTAAGCCTTTCAGTGTCGTAGCTAACGCGCTAAGTCCTCCGCCTGGGGAG
TACGGCCGCAAGGTTGAAACTCAAAGGAATTGACGGGGACCCGCACAAGCGGTGGAGCATGTGGTTTAATTCGATGCAACGCGAAGAACCT
TACCTGGTCTTGACATCCATGGAACCCTGCAGAGATGCGGGGGTGCCGTAAGGAACCATGAGACAGGTGCTGCATGGCTGTCGTCAGCTCG
TGTCGTGAGATGTTGGGTTAAGTCCCGCAACGAGCGCAACCCTTATCATTAGTTGCTACGCAAGGGCACTCTAATGAGACTGCCGGTGACA
AACCGGA
>E2499
GTAGTCCACGCCCTAAACGATGCGAACTGGAAGTTGGGGGCAACTTGGCCCTCAGTTTCGAAGCTAACGCGTTAAGTTCGCCGCCTGGGAA
GTACGGTCGCAAGACTGAAACTCAAAGGAATTGACGGGGGCCCGCACAAGCGGTGGAGTATGTGGTTTAATTCGATGCAACGCGAAGAACC
TTACCTGGCCTTGACATGTCGAGAATCCCTGAGAGATCGGGGAGTGCCTTCGGGAACTCGAACACAGGTGCTGCATGGCTGTCGTCAGCTC
GTGTCGTGAGATGTTGGGTTAAGTCCCGCAACGAGCGCAACCCTTGTCCTTAGTTGCCAGCACGTAATGGTGGGAACTCTAAGGAGACCGC
CGGTGACAAACCGGA
>E2500
GTAGTCCACGCCCTAAACGATGTCAACTGGTTGTTGGACGGCTTGCTGTTCAGTAACGAAGCTAACGCGTGAAGTTGACCGCCTGGGGAGT
ACGGCCGCAAGGTTGAAACTCAAAGGAATTGACGGGGACCCGCACAAGCGGTGGATGATGTGGTTTAATTCGATGCAACGCGAAAAACCTT
ACCTACCCTTGACATGTCAAGAATCTTGCAGAGATGTGAGAGTGCTCGAAAGAGAACTTGAACACAGGTGCTGCATGGCCGTCGTCAGCTC
GTGTCGTGAGATGTTGGGTTAAGTCCCGCAACGAGCGCAACCCTTGCCATTAGTTGCTACGAAAGGGCACTCTAATGGGACTGCCGGTGAC
AAACCGGA

woodc@DESKTOP-0NKLVFL   /c/test
$
```

查找某条序列，统计fasta文件序列条数

查找某条序列

grep 'AAAACACAGGAACCTGGGTGAAAAC' example.fa | head

```
$ grep 'AAAACACAGGAACCTGGGTGAAAAC' example.fa | head
GTAGTCCACGCCGTAAACGATGAGTGCTAGGTGTCACGGGCTTTGACCCTCGTGGTGCCGTAGCTAACGCAATAAGCACTCCGCCTGGGGA
GTACGGCCGCAAGGCTAAAACTCAAAGGAATTGACGGGGGCCCGCACAAGCGGTGGAGCATGTGGTTTAATTCGACGCGACGCGCAGAACC
TTACCTGGGCTAGAAAACACAGGAACCTGGGTGAAAACTCGGGGTGCCCTTCGGGGAATCTGTGGTTAGGTGTTGCATGGCTGTCGTCAGC
TCGTGTCGTGAGATGTTGGGTTAAGTCCCGCAACGAGCGCAACCCTTGTCGTTAGTTGCCATCATTAAGTTGGGCACTCTAACGAGACTGC
CGACCTTCAAGTCGGA
GTAGTCCACGCCGTAAACGATGGATGCTAGCCGTCGGCAAGCTTGCTTGTCGGTGGCGCAGCTAACGCATTAAGCATCCCGCCTGGGGAGT
ACGGCCGCAAGGTTAAAACTCAAAGGAATTGACGGGGGCCCGCACAAGCGGTGGAGCATGTGGTTTAATTCGATGCAACGCGAAAAACCTT
ACCTGGGCTAGAAAACACAGGAACCTGGGTGAAAACTCGGGGTGCCCTTCGGGGAATCTGTGGTTAGGTGTTGCATGGCTGTCGTCAGCTC
GTGTCGTGAGATGTTGGGTTAAGTCCCGCAACGAGCGCAACCCTTGTCGTTAGTTGCCATCATTAAGTTGGGCACTCTAACGAGACTGCCG
ACCTTCAAGTCGGA
```

统计序列条数

grep '>' example.fa | wc -l

```
$ grep '>' example.fa | wc -l
2500
```

计算fasta文件每条序列长度

统计序列长度

grep -v '>' example.fa | awk '{print length($0)}' | head

```
$ grep -v '>' example.fa | awk '{print length($0)}'| head
380
370
380
372
372
376
372
376
377
376
```

统计fasta文件序列长度分布

统计序列长度分布

```
grep -v '>' example.fa | awk '{print length($0)}' | sort | uniq -c
```

```
      7 368
      4 369
     34 370
     38 371
    569 372
     49 373
     67 374
     94 375
    271 376
    393 377
    177 378
    179 379
    289 380
     46 381
     28 382
     77 383
     26 384
     68 385
     11 386
      3 387
```

提取对应列

- # 查看样品信息表
- cat metadata.txt

- # 获取所有样品的名字
- cut -f 1 metadata.txt

```
ct586@LAPTOP-PL9JUACQ  /c/12linux
$ cat metadata.txt
SampleID        group   genotype        site
KO1     A       KO      Beijing
KO2     A       KO      Beijing
KO3     A       KO      Sanya
KO4     A       KO      Sanya
KO5     A       KO      Harbin
KO6     A       KO      Harbin
OE1     B       OE      Beijing
OE2     B       OE      Beijing
OE3     B       OE      Sanya
OE4     B       OE      Sanya
OE5     B       OE      Harbin
OE6     B       OE
```

```
ct586@LAPTOP-PL9JUACQ  /c/12linux
$ cut -f 1 metadata.txt
SampleID
KO1
KO2
KO3
KO4
KO5
KO6
OE1
OE2
OE3
OE4
OE5
OE6
```

提取对应列时略过第一行

- # 获取所有样品名字时跳过第一行
- tail -n +2 metadata.txt | cut -f 1

```
ct586@LAPTOP-PL9JUACQ  /c/12linux
$ tail -n +2 metadata.txt | cut -f 1
KO1
KO2
KO3
KO4
KO5
KO6
OE1
OE2
OE3
OE4
OE5
OE6
```

- # 对每个样品新建一个文件夹，以样品名字命名
- # 传统方式
- mkdir -p KO1 KO2 KO3 KO4

批量创建文件夹

- # 循环模式，批量创建
- for i in `tail -n +2 metadata.txt | cut -f 1`; do mkdir -p ${i}; done
- # 展示for循环中运行了什么
- for i in `tail -n +2 metadata.txt | cut -f 1`; do echo "mkdir -p ${i}"; done

- for语法格式 (红色字体不可修改)
- for i in a b c d; do sth; done

```
ct586@LAPTOP-PL9JUACQ  /c/12linux
$ for i in `tail -n +2 metadata.txt | cut -f 1`; do
 echo "mkdir -p ${i}"; done
mkdir -p KO1
mkdir -p KO2
mkdir -p KO3
mkdir -p KO4
mkdir -p KO5
mkdir -p KO6
mkdir -p OE1
mkdir -p OE2
mkdir -p OE3
mkdir -p OE4
mkdir -p OE5
mkdir -p OE6
```

sed替换文字内容

- # 假如metadata中的Beijing写错了，需要替换为Nanjing
- # s: substitute
- # /: 分隔符，可以是任意字符，前后统一就行
- # s/original/new/: 原始的替换为新的
- sed 's/Beijing/Nanjing/' metadata.txt

```
ct586@LAPTOP-PL9JUACQ  /c/12linux
$ sed 's/Beijing/Nanjing/' metadata.txt
SampleID        group   genotype        site
KO1     A       KO      Nanjing
KO2     A       KO      Nanjing
KO3     A       KO      Sanya
KO4     A       KO      Sanya
KO5     A       KO      Harbin
KO6     A       KO      Harbin
OE1     B       OE      Nanjing
OE2     B       OE      Nanjing
OE3     B       OE      Sanya
OE4     B       OE      Sanya
OE5     B       OE      Harbin
OE6     B       OE      Harbin
```

awk 提取两列并交换顺序(成为高手必学)

- # 取出metadata.txt的前两列，并把第二列作为输出结果的第一列
- # awk擅长于对文件按行操作，每次读取一行，然后进行相应的操作。
- # awk读取单个文件时的基本语法格式是awk 'BEGIN{OFS=FS="\t"}{print $0, $1;}' filename。
- # 读取多个文件时的语法是awk 'BEGIN{OFS=FS="\t"}ARGIND==1{print $0, $1;}ARGIND==2{}' file1 file2。
- # awk后面的命令部分是用引号括起来的，可以单引号，可以双引号，但注意不能与内部命令中用到的引号相同，否则会导致最相邻的引号视为一组，引发解释错误。
- # OFS: 文件输出时的列分隔符 (output field separator) # FS: 文件输入时的列分隔符 (field separator)
- # BEGIN: 设置初始参数，初始化变量 # END: 读完文件后做最终的处理
- # 其它{}：循环读取文件的每一行,大括号内的命令对每一行都有效，除非有额外判断
- # $0表示一行内容；$1, $2, … $NF表示第一列，第二列到最后一列。
- # NF (number of fields)文件多少列；NR (number of rows) 文件读了多少行
- # FNR 当前文件读了多少行，常用于多文件操作时
- # a[$1]=1: 索引操作，类似于python中的字典，在ID map、注释和统计中有很多应用
- # 提取前两列并交换位置
- awk 'BEGIN{OFS=FS="\t"}{print $2,$1}' metadata.txt

```
ct586@LAPTOP-PL9JUACQ  /c
$ awk 'BEGIN{OFS=FS="\t"}
t
group   SampleID
A       KO1
A       KO2
A       KO3
A       KO4
A       KO5
A       KO6
B       OE1
B       OE2
B       OE3
B       OE4
B       OE5
B       OE6
```

awk计算每个样品的生物重复个数

- # 计算每个样品的重复的个数
- awk 'BEGIN{OFS=FS="\t"}{a[$3]+=1}END{for(i in a) print i,a[i];}' metadata.txt

```
ct586@LAPTOP-PL9JUACQ  /c/12linux
$ awk 'BEGIN{OFS=FS="\t"}{a[$3]+=1}END{for(i in a)
print i,a[i];}' metadata.txt
WT      6
OE      6
genotype        1
KO      6
```

- # 结果多了一行，略过标题行
- awk 'BEGIN{OFS=FS="\t"}{if(FNR>1) a[$3]+=1}END{for(i in a) print i,a[i];}' metadata.txt

```
ct586@LAPTOP-PL9JUACQ  /c/12linux
$ awk 'BEGIN{OFS=FS="\t"}{if(FNR>1) a[$3]+=1}END{fo
r(i in a) print i,a[i];}' metadata.txt
WT      6
OE      6
KO      6
```

awk中常用几个符号

- =： 一个等号表示赋值
- ==：表示判断两侧的变量是否相等，如FNR==1，若相等返回True
- !=：表示判断两侧的变量是否不相等，若不等，返回True
- >, <：判断数值或字符串的大小
- +=：自加操作 a+=1 等同于 a = a+1
- %：取余数
- +, -, * , / ：加减乘除
- &&, ||, !：逻辑与，逻辑或，逻辑非

awk中的括号和引号

- ()：一般用于函数中，传递参数
- []：一般用于索引列表或字典，取出列表中第几位元素，或字典中某个key对应的value
- {}：代码块，多行语句放在一起，属于一个层级
- ""，''：双引号和单引号括起的是字符串，bash中双引号中的变量可以解析，单引号不可以。同样的引号不可以嵌套，如 awk "{print "ehbio"}"是不对的，要写成awk '{print "ehbio"}'或awk "{print 'ehbio'}"。
- ;：分号用于分割语句块

Shell程序中的其它符号

- \t：代表TAB键
- *：代表任意字符
- |：管道符，传递数据，上一条命令的输出作为下一条命令的输入
- >：输出重定向，常用语把输出结果写入文件

Linux常用命令小结

pwd	显示当前目录	cat	查看文件	head	筛选文件开头N行
ls	列出目录内容	ping	测试网络连接	tail	筛选文件结尾N行
ls -l	详细列出目录内容	less -S	上下翻页查看文件	grep	筛选特定关键词的行
cd	切换当前目录	cp	复制文件	cut	列操作
mkdir	创建目录	mv	移动文件	sed	文本替换
chmod	修改文件权限	rm	删除文件	sort	排序
gunzip	解压缩gz文件	awk	文本处理工具	uniq -c	去冗余并统计
|	管道，命令串联	Ctrl+D	终止编辑	Ctrl+C	终止运行程序

系统学习材料推荐

Unix/Linux 命令参考 FOSSwire.com

文件命令

ls – 列出目录
ls -al – 使用格式化列出隐藏文件
cd *dir* - 更改目录到 *dir*
cd – 更改到 home 目录
pwd – 显示当前目录
mkdir *dir* – 创建目录 *dir*
rm *file* – 删除 *file*
rm -r *dir* – 删除目录 *dir*
rm -f *file* – 强制删除 *file*
rm -rf *dir* – 强制删除目录 *dir**
cp *file1 file2* – 将 *file1* 复制到 *file2*
cp -r *dir1 dir2* – 将 *dir1* 复制到 *dir2*; 如果 *dir2* 不存在则创建它
mv *file1 file2* – 将 *file1* 重命名或移动到 *file2*; 如果 *file2* 是一个存在的目录则将 *file1* 移动到目录 *file2* 中
ln -s *file link* – 创建 *file* 的符号连接 *link*
touch *file* – 创建 *file*
cat > *file* – 将标准输入添加到 *file*
more *file* – 查看 *file* 的内容
head *file* – 查看 *file* 的前 10 行
tail *file* – 查看 *file* 的后 10 行
tail -f *file* – 从后 10 行开始查看 *file* 的内容

进程管理

ps – 显示当前的活动进程
top – 显示所有正在运行的进程
kill *pid* – 杀掉进程 id *pid*
killall *proc* – 杀掉所有名为 *proc* 的进程 *
bg – 列出已停止或后台的作业
fg – 将最近的作业带到前台
fg *n* – 将作业 *n* 带到前台

文件权限

chmod *octal file* – 更改 *file* 的权限
- 4 – 读 (r)
- 2 – 写 (w)
- 1 – 执行 (x)

示例:
chmod 777 – 为所有用户添加读、写、执行权限
chmod 755 – 为所有者添加 rwx 权限, 为组和其他用户添加 rx 权限
更多选项参阅 man chmod.

SSH

ssh *user@host* – 以 *user* 用户身份连接到 *host*
ssh -p *port user@host* – 在端口 *port* 以 *user* 用户身份连接到 *host*
ssh-copy-id *user@host* – 将密钥添加到 *host* 以实现无密码登录

搜索

grep *pattern files* – 搜索 *files* 中匹配 *pattern* 的内容
grep -r *pattern dir* – 递归搜索 *dir* 中匹配 *pattern* 的内容
command | grep *pattern* – 搜索 *command* 输出中匹配 *pattern* 的内容

系统信息

date – 显示当前日期和时间
cal – 显示当月的日历
uptime – 显示系统从开机到现在所运行的时间
w – 显示登录的用户
whoami – 查看你的当前用户名
finger *user* – 显示 *user* 的相关信息
uname -a – 显示内核信息
cat /proc/cpuinfo – 查看 cpu 信息
cat /proc/meminfo – 查看内存信息
man *command* – 显示 *command* 的说明手册
df – 显示磁盘占用情况
du – 显示目录空间占用情况
free – 显示内存及交换区占用情况

压缩

tar cf *file.tar files* – 创建包含 *files* 的 tar 文件 *file.tar*
tar xf *file.tar* – 从 *file.tar* 提取文件
tar czf *file.tar.gz files* – 使用 Gzip 压缩创建 tar 文件
tar xzf *file.tar.gz* – 使用 Gzip 提取 tar 文件
tar cjf *file.tar.bz2* – 使用 Bzip2 压缩创建 tar 文件
tar xjf *file.tar.bz2* – 使用 Bzip2 提取 tar 文件
gzip *file* – 压缩 *file* 并重命名为 *file.gz*
gzip -d *file.gz* – 将 *file.gz* 解压缩为 *file*

网络

ping *host* – ping *host* 并输出结果
whois *domain* – 获取 *domain* 的 whois 信息
dig *domain* – 获取 *domain* 的 DNS 信息
dig -x *host* – 逆向查询 *host*
wget *file* – 下载 *file*
wget -c *file* – 断点续传

安装

从源代码安装:
./configure
make
make install
dpkg -i *pkg.deb* – 安装包 (Debian)
rpm -Uvh *pkg.rpm* – 安装包 (RPM)

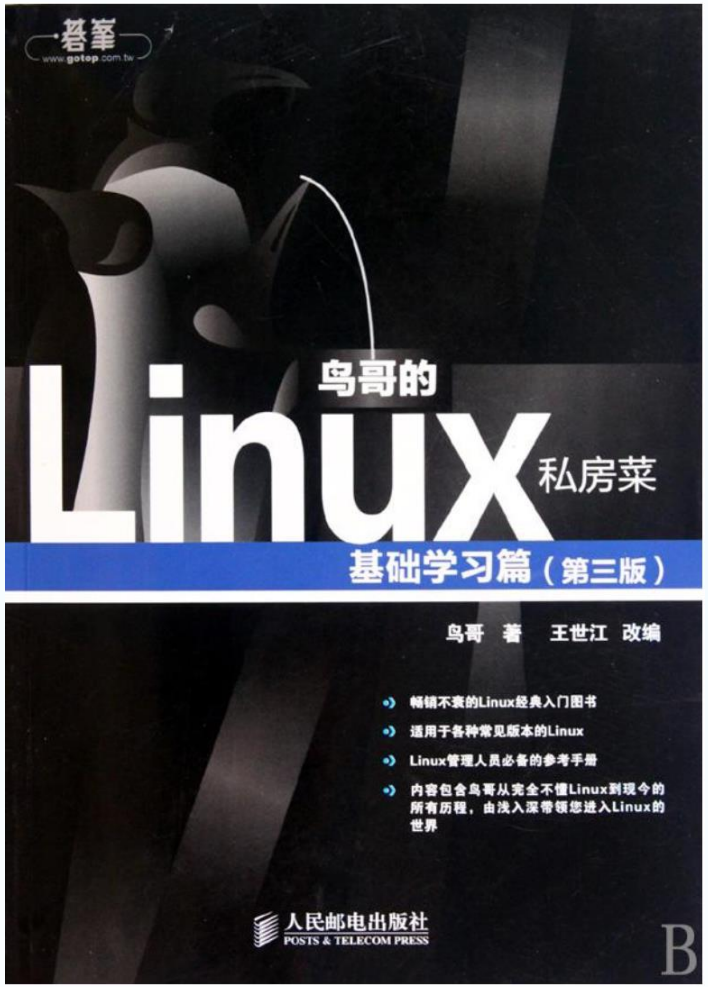
快捷键

Ctrl+C – 停止当前命令
Ctrl+Z – 停止当前命令, 并使用 fg 恢复
Ctrl+D – 注销当前会话, 与 exit 相似
Ctrl+W – 删除当前行中的字
Ctrl+U – 删除整行
!! - 重复上次的命令
exit – 注销当前会话

* 小心使用。

翻译/Toy <http://LinuxTOY.org>

https://fosswire.com/post/2007/08/unixlinux-command-cheat-sheet/



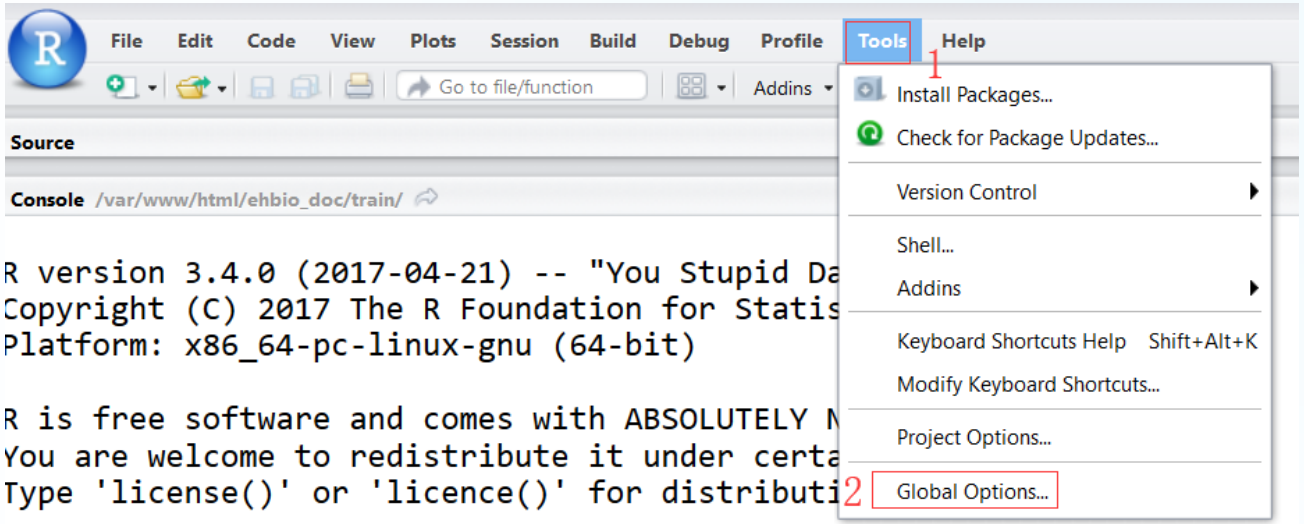
Linux

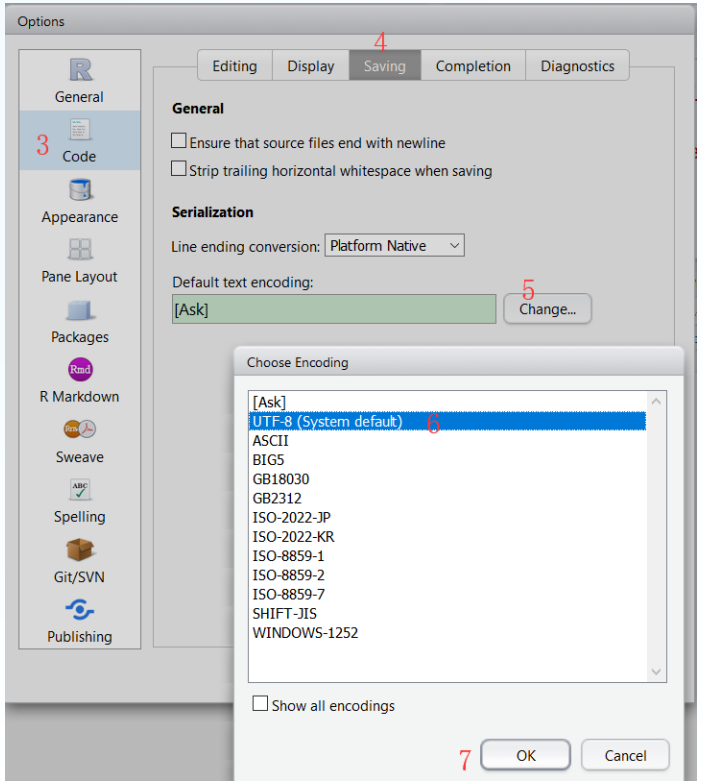
- Linux - 总目录
- Linux - 文件和目录
- Linux - 文件操作
- Linux - 文件内容操作
- Linux - 环境变量和可执行属性
- Linux - 管道、标准输入输出
- Linux - 命令运行监测和软件安装
- Linux - 常见错误和快捷操作
- Linux - 文件列太多，很难识别想要的信息在哪列；别焦急，看这里。
- Linux - 文件排序和FASTA文件操作
- Linux - 应用Docker安装软件
- Linux - Conda软件安装方法
- Linux - 服务器数据定期同步和备份方式
- Linux - VIM的强大文本处理方法
- Linux - 查看服务器配置信息
- Linux - SED操作，awk的姊妹篇
- Linux - 常用和不太常用的实用awk命令
- Linux - 那些查找命令
- Linux - 原来你是这样的软连接
- Bash概论 - Linux系列教程补充篇

生信宝典Linux系列教程 http://mp.weixin.qq.com/s/i71OMaUu6QtcY0pt1njHQA

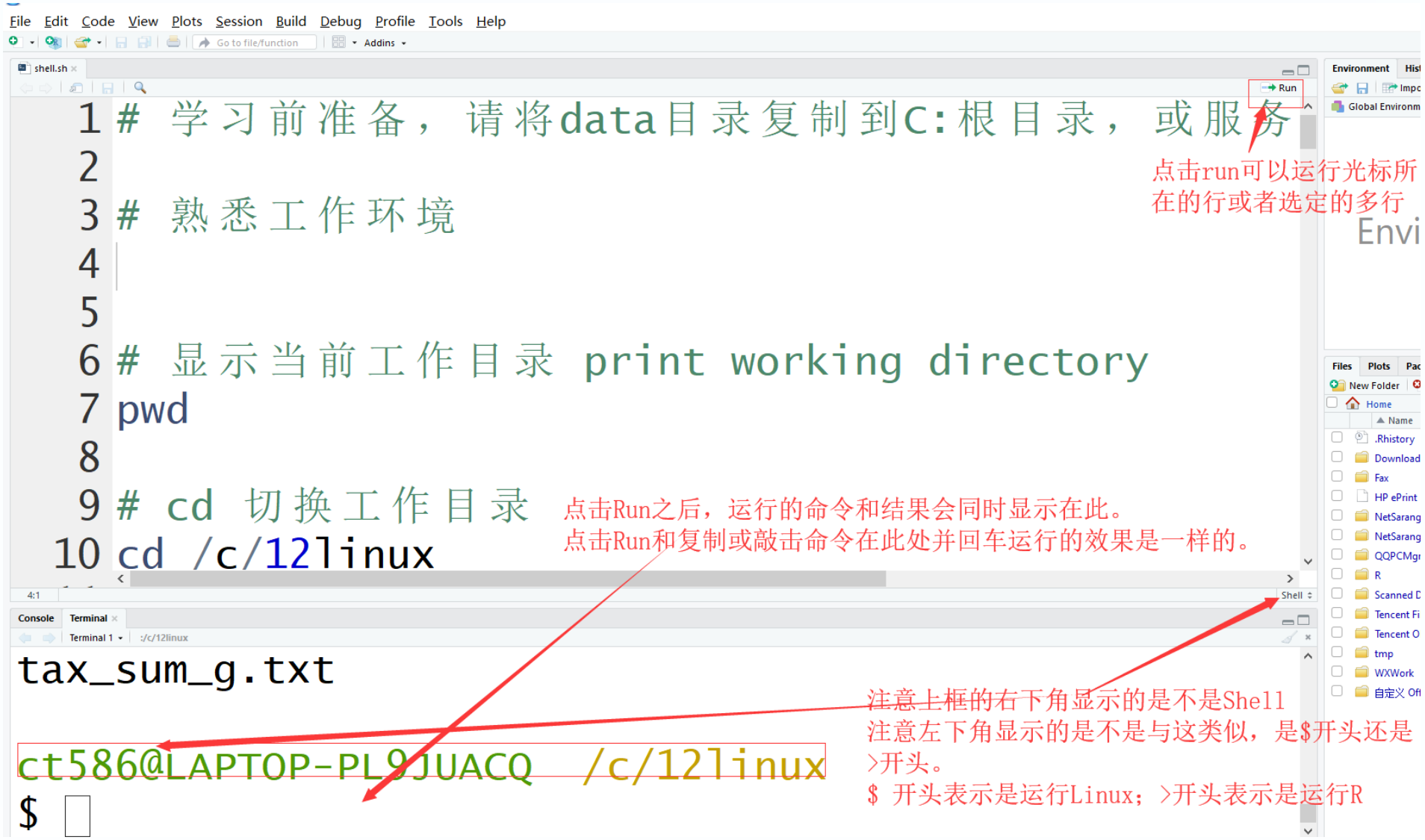
实际操作 – 打开文件和乱码解决

- 拷贝12Linux文件夹到C盘amplicon目录下 （任意一个盘任意不含中文和空格的目录都可以，这里为了方便统一都放在C盘amplicon下）。
- File – Open file – 打开shell.sh；若出现乱码，按图示操作。





实际操作 – 注意下面几个关键点



实际操作 – 文件路径

```
ct586@LAPTOP-PL9JUACQ   /c/12linux   1
$ cd test

ct586@LAPTOP-PL9JUACQ   /c/12linux/test   2
$ ls # 查看文件夹内容
test.txt

                                注意目录的变化

ct586@LAPTOP-PL9JUACQ   /c/12linux/test
$ # 切换至上级目录

ct586@LAPTOP-PL9JUACQ   /c/12linux/test
$ cd ..

ct586@LAPTOP-PL9JUACQ   /c/12linux   3
$
```

开始接触命令行，一个难跨越的概念是文件路径。Linux系统虽然好用，但还没智能到只给文件名就能判断路径的地步，实际也没必要，而且会引发危险。在Windows下访问文件时，会一层层打开文件夹去查看，Linux下也类似，只不过用cd代替了打开操作。如果碰到文件找不到的错误，一定先看下当前所在目录和文件所在目录。

扫码关注生信宝典，学习更多生信知识

扫码关注宏基因组，获取专业学习资料

易生信，没有难学的生信知识