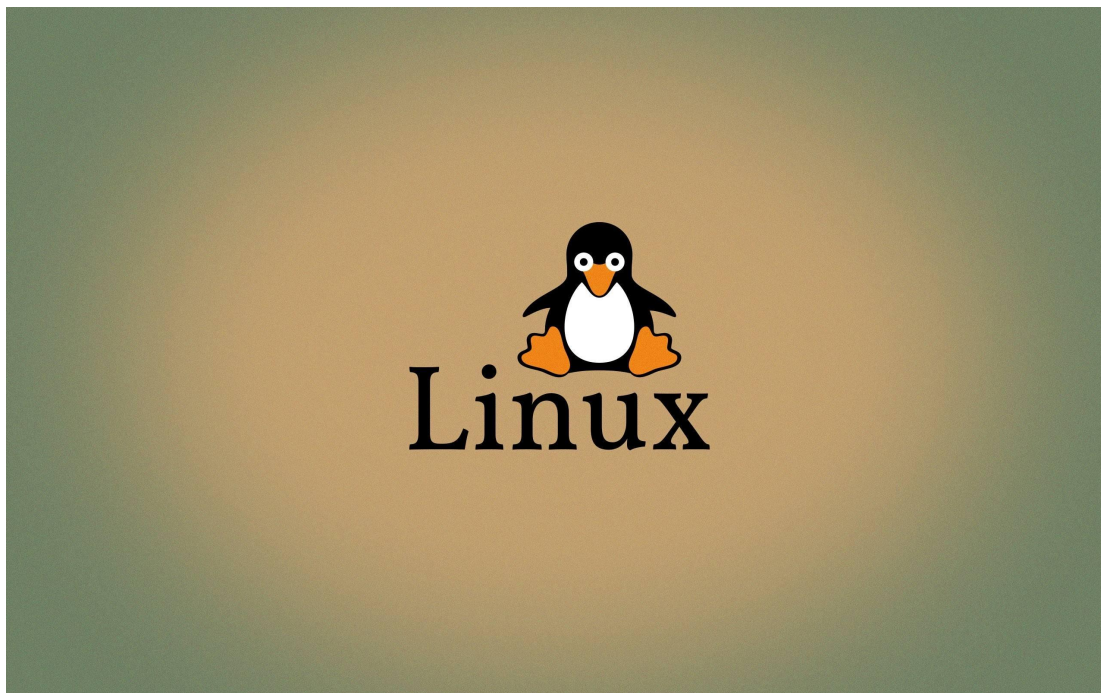


生物信息常用的 linux 命令大全



生物信息学需要从指数增长的生物学数据中，挖掘大量数据中所包含的生物学意义，对计算资源的要求较高。我们很多时候需要在集群上分析数据，而 Linux 是最常用的服务器操作系统。因此对于生物信息人员而言，Linux 学习是必须的。

在实际运用过程中，我们通常根据需要，通过管道符“|”同时使用好几种 Linux 命令，避免花费大量时间找软件、写代码，有些情况下可以代替 Excel 等办公软件。如下就是一些 linux 命令的高级使用实例。

如下是一个宏基因组的 NR 注释结果示例文件。

```
cmd: head -5 nr.taxonomy.abundance.xls
Taxon B1 B2 C1 C2 E1 E2
Unassigned; Unassigned; Unassigned; Unassigned; Unassigned; Unassigned; Unassigned; 10.0 7.0 3.0 5.0 11.0 0.0
k_Archaea; p_Candidatus_Aenigmarchaeota; c_Unclassified; o_Unclassified; f_Unclassified; g_Unclassified; s_Candidatus_Aenigmarchaeota; archaeon 0.0 0.0 0.0 0.0 0.0 3.0
k_Archaea; p_Candidatus_Verstraetearchaeota; c_Unclassified; o_Unclassified; f_Unclassified; g_Unclassified; s_Candidatus_Verstraetearchaeota; archaeon 0.0 0.0 0.0 0.0 1.0 0.0
k_Archaea; p_Crenarchaeota; c_Unclassified; o_Unclassified; f_Unclassified; g_Unclassified; s_uncultured_crenarchaeote 11.0 5.0 0.0 0.0 0.0 0.0
```

`grep -v '^Taxon' nr.taxonomy.abundance.xls | cut -d ";" -f 7- | head -5 #从`

中筛选种级别的注释结果，并展示前 5 行。

```
cmd: grep -v '^Taxon' nr.taxonomy.abundance.xls | cut -d ";" -f 7- | head -5
Unassigned      10.0    7.0    3.0    5.0    11.0    0.0
s_Candidatus_Aenigmarchaeota_archaeon 0.0    0.0    0.0    0.0    0.0    3.0
s_Candidatus_Verstraetearchaeota_archaeon 0.0    0.0    0.0    0.0    0.0    1.0    0.0
s_uncultured_crenarchaeote      11.0    5.0    0.0    0.0    4.0    1.0
s_uncultured_crenarchaeote_29d5    0.0    0.0    0.0    0.0    1.0    0.0
```

`cut -d ";" -f 6- nr.taxonomy.abundance.xls | grep "Nitrososphaera" | head #`

从中筛选"Nitrososphaera"属的注释结果，并展示前 5 行。

```
cmd: cut -d ";" -f 6- nr.taxonomy.abundance.xls | grep "Nitrososphaera" | head
g_Nitrososphaera; s_Candidatus_Nitrososphaera_evergladensis 0.0    6.0    0.0    0.0    0.0    0.0
g_Nitrososphaera; s_Candidatus_Nitrososphaera_gargensis 3.0    5.0    0.0    0.0    2.0    0.0
g_Nitrososphaera; s_Candidatus_Nitrososphaera_sp._13_1_40CM_48_12 2.0    0.0    0.0    0.0    1.0    0.0
g_Nitrososphaera; s_Nitrososphaera_viennensis 12.0    2.0    2.0    0.0    4.0    6.0
```

`cut -d ";" -f 7- nr.taxonomy.abundance.xls | awk -F "\t" 'NR>1{sum =`

`$2+$3+$4+$5+$6+$7;print $1"\t"sum}' | head -5 #计算各个种在所有样`

本种的丰度之和，展示前 5 行。

```
cmd: cut -d ";" -f 7- nr.taxonomy.abundance.xls | awk -F "\t" 'NR>1{sum = $2+$3+$4+$5+$6+$7;print $1"\t"sum}' | head -5
Unassigned      36
s_Candidatus_Aenigmarchaeota_archaeon 3
s_Candidatus_Verstraetearchaeota_archaeon 1
s_uncultured_crenarchaeote 21
s_uncultured_crenarchaeote_29d5 1
```

Linux 下的所有操作都是通过众多的命令行实现的。我们只需要掌握如下的常用命令的常用方法，就能熟练使用 Linux 系统了。大家可以根据下面的说明文档，在 Linux 上多次练习，从而融会贯通的使用 Linux 的众多命令。

1 文件和目录管理

1.1 管理目录

1.1.1 cd

命令说明：切换目录。Linux 文件系统的目录结构是树状级结构图。

`cd`（change directory）命令，可以将目录，从当前目录切换至目标目录。

语法：`cd <dirname>`

实例：cd /path/dirname #切换到指定的/path/dirname 目录

cd ~ #切换到家目录

cd ../ #切换到上一层目录

cd / #切换到根目录

1.1.2 ls 和 ll

命令说明：ls 和 ll 都可以列出目录下的文件、子目录。ls 只列出文件名或目录名。而 ll 是 ls -l 的别名，会列出该目录下的文件或目录的详细信息。

语法：ls [-alrtAFR] <dirname/filename>

常用参数：-l 除文件名外，列出文件类型、权限、所有者、大小等详细信息。

-h 文件大小的单位由字节 Byte，变换为“K”、“M”、“G”、“T”（需要与-l 同时使用）。

-a 显示“.”开头的隐藏文件。

-t 文件按照时间顺序，越新的越前面。

1.1.3 mkdir

命令说明：创建文件夹（目录）。

语法：mkdir [-p] <dirname>

常用参数：-p 如果需要创建目录的父目录（上层目录）不存在，就一并创建父目录。

实例：mkdir -p /new_dirname/test_dir #创建目录/new_dirname/test_dir（当 new_dirname 不存在的时候，一并创建）。

1.1.4 pwd

命令说明：显示当前工作目录的绝对路径。含有“.”、“..”和“~”等的路径为相对路径，只由目录名称和“/”组成的为绝对路径。

语法：pwd

1.2 处理文件

1.2.1 创建文件

1.2.1.1 touch

命令说明：创建一个空的新文件。

语法：touch <filename>

1.2.1.2 vi

命令说明：vi 有三种状态，分别是命令模式（command mode）、插入模式（Insert mode）和底行模式（last line mode）。在插入模式下，我们才可以输入文字。然后在底行模式下，将文件保存。

语法：vi <filename>

1.2.2 文件的复制、移动、链接、删除

1.2.2.1 cp

命令说明：复制文件或目录，相当于 Windows 系统下的 copy。

语法：cp <source dirname/filename> <target dirname/filename> [-r]

常用参数：-r 将源目录及其子目录下的所有文件，复制到目标目录下（复制目录的时候，必须使用）。

实例：cp file_a dir_name #将文件 file_a 拷贝到目录 dir_name 下。

`cp file_a file_b` #将文件 `file_a` 复制并重新命名为 `file_b`。

`cp -r dir_a dir_b` #将目录 `dir_a` 及其子目录下的所有文件，拷贝到目录 `dir_b` 下。

1.2.2.2 mv

命令说明：用于将文件移动到目标目录下；重命名文件或目录；移动目录到目标目录下。

语法：`mv <source filename/> <target dirname/>`

实例：`mv file_a file_b` #将目录 `file_a` 重命名为 `file_b`。

`mv dir_a dir_b` #如果目录 `dir_b` 不存在，则将 `dir_a` 重命名为 `dir_b`；如果目录 `dir_b` 存在，则将 `dir_a` 移动到 `dir_b` 下。

`mv file1 file2 dirname1 test_dirname` #将文件 `file1` 和 `file2`、目录 `dirname1`，移动到 `test_dirname` 下。

1.2.2.3 ln

命令说明：为某个文件在另一个位置建立一个同步的链接。当我们需要在不同目录下使用同一个文件或目录的时候，不需要在不同目录下重复保存该文件。我们可以将文件放在某个目录下，在其他目录下，链接（link）该文件，避免重复占用磁盘空间，减少资源浪费。

语法：`ln <source dirname/> <target dirname/> [-s]`

常用参数：`-s` 默认创建硬链接。使用参数`-s`，则创建软链接。软链接，以路径形式存在，相当于 Windows 系统的快捷方式。硬链接，以文件副本形式存在，实际不占用内存（和源文件）指向同一个数据块。删除源文件后，软链接失效，但是硬链接不受影响。

实例：ln file_a file_b #对源文件 file_a，建立一个硬链接 file_b。

ln -s file_a file_b #对源文件 file_a，建立一个软链接 file_b。

1.2.2.4 rm

命令说明：删除文件或目录。

语法：rm [-rif] <dirname/filename>

常用参数：-r 将目录及其子目录，逐一删除。对目录执行删除操作，必须加上-r。

-i 在执行删除操作之前，会进行逐一询问。

-f 不询问，直接删除，相当于强制删除。

实例：rm -r dirname1 filename1 #删除目录 dirname1、文件 filename1。

1.2.3 文件内容查看

1.2.3.1 cat

命令说明：将文件内容显示到标准输出（或使用管道命令“|”传递作为下一个命令的标准输入，或使用“>”重定向到其他文件）。

语法：cat [-n] <file1 file2 ...> [> new_file]

常用参数：-n 在每行内容之前，加上行号显示。

实例：cat -n file #将 file 的内容加行号，显示在屏幕上。

cat file1 file2 > new_file #将 file1 和 file2 的内容，重定向到 new_file 中。（file1 的内容在前，file2 的内容在后。）

cat file1 file2 >> new_file #将 file1 和 file2 的内容，依次追加到 new_file 后。

cat file | wc -l #使用 cat 查看 file 内容，再传递到 wc -l 计算文件的行

数。

1.2.3.2 less

命令说明： 查看文件的内容。

语法： less [-SN] <file>

常用参数： -S 当每行长度长于屏幕宽度的时候，默认显示在下一行，使用-S 则不显示在下一行。

-N 显示行号。

实例： less -SN #使用 less 打开文件后，可以使用“Page Up”向上翻页，使用“Page Down”向下翻页，使用“/pattern”向下匹配。

1.2.3.3 head

命令说明： 显示文件开头部分的内容。

语法： head [-n] <file>

常用参数： -n <number> 显示文件前 number 行的内容（默认显示前 10 行）。

实例： head -n 20 file #显示文件前 20 行的内容。

1.2.3.4 tail

命令说明： 显示文件末尾部分的内容。

语法： tail [-n] <file>

常用参数： -n <number> 显示文件末尾 number 行的内容（默认显示末尾 10 行）。

实例： tail -n 20 file #显示文件末尾 20 行的内容。

1.2.3.5 cut

命令说明：显示文件的每一行指定范围内的字符。

语法：cut [-cdf] <file>

常用参数：-c 以字符为单位进行分割。-c num1-num2 意为显示文件的每一行的第 num1 到 num2 的字符。

-d 自定义域分割符，默认 tab 键“\t”。

-f num1-num2 显示域分割符分割后，每行第 num1 到 num2 的域。

实例：cut -c 1-5 file #显示文件的每一行的前 5 个字符。

cut -d “ ” -f 1-5 file #显示以空格分割后的，文件的前 5 列。

1.2.4 文件压缩

1.2.4.1 gzip

命令说明：压缩文件或目录，生成以“.gz”结尾的压缩文件。

语法：gzip [-r][-1...9] <filename/dirname>

常用参数：-r 对目录下的文件进行压缩，但目录本身不压缩。

-1...9 数值越大，压缩效率越高（默认 6），压缩速度越慢，生成的压缩文件越小。

实例：gzip file1 #对源文件 file1 进行压缩，生成 file1.gz，并删除源文件 file1。

解压命令：使用 gunzip 或 gzip -d，例如 gunzip file1.gz。

1.2.4.2 zip

命令说明：压缩文件或目录，生成“.zip”结尾的压缩文件。

语法：zip [-r][-1...9] <filename/dirname>

常用参数：-r 对目录下的所有子目录、文件一起压缩。

-1...9 数值越大，压缩效率越高（默认 6），压缩速度越慢，生成的压缩文件越小。

实例：`zip -r new.zip dirname1` #将目录 `dirname1` 压缩到 `new.zip` 压缩文件内。

解压命令：`unzip`，例如 `unzip new.zip`。

1.2.4.3 tar

命令说明：Linux 下最常用的打包程序，生成的包通常称为 tar 包（以“.tar”结尾）。可以在 tar 包的基础上，使用压缩命令，进一步压缩处理。

语法：`tar [-cxzvf] <tar_file>`

常用参数：-f 指定 tar 文件，后面需要立即接 tar 文件名，不能再接参数。

-c 创建一个 tar 文件。

-x 从 tar 文件中，获取源目录和源文件，相当于打开 tar 包。

-z 和-c 同用，调用 `gzip` 来压缩 tar 文件；和-x 同用，调用 `gunzip` 来解压缩 tar 文件。

-v 显示命令执行的过程。

实例：`tar -cf test.tar file1 dir1` #将文件 `file1`、目录 `dir1` 打包到 `test.tar` 中。

`tar -xf test.tar` #提取 `test.tar` 中的文件。

`tar -czvf test.tar file1 dir1` #将文件 `file1`、目录 `dir1` 打包并压缩为 `test.tar.gz`，显示详细过程。

`tar -xzvf test.tar file1 dir1` #将 test.tar.gz 解压缩，提取其中文件。

1.2.5 文件中的数据统计与处理

1.2.5.1 sort

命令说明：将文本文件中的内容按一定顺序进行排序。

语法：`sort [-nk] <filename>`

常用参数：`-n` 文件内容按照数值从小到大排序（默认按照 `ascii` 码从小到大排序）。

`-k num` 按照第 `num` 列的内容对文本文件进行排序（默认按照第一列）。

实例：`sort -n -k2 file.txt` #文件 `file.tx` 按照第二列的数值从小到大排序并输出（默认打印在屏幕上）。

1.2.5.2 uniq

命令说明：合并文件中相邻的相同行。使用 `sort` 使文件排序后，再用 `uniq` 就可以去除文件中的重复行。

语法：`uniq [-cd] <filename>`

常用参数：`-c` 在输出的第一列显示该行的重复次数。

`-d` 只显示有重复的行。

实例：`sort file | uniq -d` #对 `file` 进行排序后，显示重复行。

`sort file | uniq > out.file` #将 `file` 中的所有行去除重复后，输出到 `out.file`。

1.2.5.3 wc

命令说明：统计文本文件中的字节数、字数、行数。

语法：`wc [-cwl] <file>`

常用参数：`-c` 统计文件中的字节数。

-w 统计文件中的字数（单词数目）。

-l 统计文件中的行数。

实例：sort file | wc -l #统计文件中的非重复行的行数。

1.2.5.4 grep

命令说明：在文件中查找符合条件的行（含有、不含有指定字符串）。

语法：grep [-niv] <string> <filename>

常用参数：-n 输出的第一列显示行数。

-i 查找 file 中含 string 的行的时候，忽略 string 的大小写。

-v 不匹配，输出不含 string 的行。

实例：grep -nv hello file #输出文件中不含“hello”的行，并显示行数。

1.2.5.5 sed

命令说明：对文本文件进行处理（可以编辑文件）。

语法：sed [-in] 'command' <filename>

常用参数：-i 直接修改原文件（默认修改后输出到屏幕，原文件不改动）。

-n 只显示匹配处理的行。

实例：sed 's/world/hello/' file #将 file 的所有行中第一个“world”换为“hello”在屏幕输出。

sed 's/world/hello/g' file #将 file 的所有行中所有“world”换为“hello”在屏幕输出。

sed -i's/world/hello/g' file #将 file 的所有行中所有“world”修改为“hello”，file 被改动。

`sed -n'2,3p'file` #输出 file 中的第 2 到 3 行。（指定文件中的某些行输出）

1.2.5.6 awk

命令说明：Linux 中最强大的数据处理工具之一，功能强大。一些情况下，我们无需编程，使用 `awk` 就可以得到理想结果。

语法：`awk [-F] '{[pattern] action}' <filename>`

常用参数：`-F` 指定列的分隔符，默认是空白（空格、`tab` 键）。

实例：`awk -F ":" '{sum += $1}END{print sum}' file` #以“:”为分割符，对 file 的第一列进行加和，输出第一列的总和。

`awk 'NR>=10{print $0}' file` #输出文件的第 10、11、12.....行。

1.2.6 特殊的文件处理命令

1.2.6.1 chmod 设置权限

命令说明：设置文件、目录的权限。Linux 下文件调用权限分为三级：文件所有者（`owner`）、用户组（`group`）、其他用户（`other users`）。权限分为 `r`（是否可读）、`w`（是否可写）和 `x`（是否可执行）。例如“`drwxr-xr-x`”中，“`d`”代表为目录（文件夹），“`rw`”代表文件所有者可以读、写、执行该目录，`group` 和 `other users` 的“`r-x`”代表他们只能读、执行该目录。`mode` 分为符号模式（`r`、`w`、`x`）和绝对模式（八进制数字模型）。其中，1 代表“`x`”、2 代表“`w`”、4 代表“`r`”，数字可以组合使用，例如 5 代表“`xr`”、7 代表“`xwr`”。

语法：`chmod [-R mode] <filename/dirname>`

常用参数：`-R` 对目录及目录下的所有文件，均设置权限。

实例：`chmod +x file` #对文件所有者（owner）、用户组（group）、其他用户（other users）都增加执行 file 的权限。（当 file 是脚本、软件的时候，非常实用。）

`chmod -R 777 dirname` #使得 owner、group 和 other users 都可以读、写、执行目录 dirname 下的所有文件。（当报错，需要求助伙伴查找问题的时候，很好用。使用时候需要谨慎，避免文件被意外更改。）

1.2.6.2 du 显示目录大小

命令说明：du 可以显示目录、文件的大小。相比于 `ll -h` 显示文件大小，优势在于可以统计目录下所有文件的大小之和。

语法：`du [-sh] <filename/dirname>`

常用参数：-s 只显示总计大小。

-h 大小的单位由字节 Byte，变换为“K”、“M”、“G”、“T”。

实例：`du -sh dirname` #显示目录 dirname 下的所有子目录、文件的大小，以“K”、“M”、“G”、“T”为单位显示。

1.2.6.3 md5sum 验证文件

命令说明：生成文件的 md5 码，用于核实数据传输是否正确。一般情况下，md5 码一致，则代表文件一致。

语法：`md5sum [-c] <filename>`

常用参数：-c 校验 md5 值是否与文件一致，文件传输是否正确。

实例：`md5sum file > md5.txt` #生成 file 的 md5 码，重定向输出到 md5.txt 文件中。

`md5sum -c md5.txt` #核实 md5.txt 文件中的 md5 码，与 file 的 md5 码

是否一致。

1.2.6.4 find 查找文件

命令说明：在目录下，查找目标目录或文件。

语法：find <path> [expression]

常用参数：path 在 path 目录下寻找。

expression 查找模式，常用的有 -name <filename> 按照文件名寻找，
-type <l/d/f> 按照文件类型寻找（l 指符号链接文件，f 指普通文件，d 指目录文件）。

实例：find /test/path -name “*txt” -type f # 在 /test/path 及其子目录下寻找以 txt 结尾的普通文件。

2 进程管理

2.1 进程查看

2.1.1 jobs

命令说明：显示当前节点、当前窗口的后台任务。

语法：jobs [-l]

常用参数：-l 显示任务进程 ID。

实例：jobs 和 jobs -l

2.1.2 ps

命令说明：显示用户当前所有进程的快照（所有节点）。

语法：ps [-xf]

常用参数：-x 显示没有控制终端的进程，同时显示各个命令的路径。

-f 全部列出，并显示进程之间的关系。

实例：ps xf

2.2 进程控制

2.2.1 kill

命令说明：删除或者控制执行中的进程。

语法：kill [-num] <PID>

常用参数：-num 通过 num 传递信号，控制进程。（默认 15，终止进程）

实例：kill 1990 #终止进程号为 1990 的进程。

kill %1 #终止 job number 为 1 的任务（jobs 命令查看 job number）。

2.2.2 bg

命令说明：将前台进程转为后台（需要先使用 ctrl+z 暂停任务）；继续后台暂停任务。

语法：bg [%number]

常用参数：%number 将编号 number 的后台暂停任务继续执行，number 由 jobs 命令得到。

实例：ctrl+z, bg #将前台的任务暂停后，转为后台。

bg %8 #将后台编号 8 的暂停任务，继续执行。

2.2.3 fg

命令说明：将后台任务转为前台。

语法：fg [%number]

常用参数：%number 将编号为 number 的任务从后台转为前台。

实例：`fg` #将最近的一个后台任务转为前台。

`fg %8` #将编号 8 的任务转为前台执行。

2.2.4 转后台&

命令说明：加在命令后面，使得命令在后台运行。

实例：`sh command &` #后台执行 `command` 命令。

2.2.5 nohup

命令说明：命令前面加上 `nohup`，则用户退出登录状态，任务也正常执行。常与转后台“&”一起使用，使任务在后台执行，退出后仍然正常运行。

实例：`nohup sh test.sh &` #投递任务 `test.sh`，日志默认输出到 `nohup.out` 文件。

`nohup sh test.sh >out 2>err&` #投递任务 `test.sh`，标准输出输出至 `out` 文件，标准错误输出至 `err` 文件。