

---

“Linux 生物信息基础”课程  
小组集体练习、讨论、交流

## 总 结 报 告

组：4    次：1    组长：陈奕晗    执笔：邹济平

1. 时间：2021 年 4 月 1 日，14:10 ~ 17:00
2. 地点：王克桢 348
3. 人员：陈奕晗、邹济平、朱瑾煜、高培翔
4. 方式：线下讨论
5. 主题：Linux 指令复习，文本编辑器 nano 的使用复习及 EMBOSS 序列比对复习

### 6. 内容

#### 6.1 已学 Linux 指令复习总结

##### 1) head

显示文件的前面一部分，默认为前 10 行。

head -n 3 或 -n +3            显示前 3 行  
head -n -3                    显示除前 3 行外的其他行

##### 2) tail

显示文件的后面一部分，默认为后 10 行

tail -n 3 或 -n -3            显示后 3 行  
tail -n +3                    显示除后 3 行外的其他行

##### 3) sort

按行排序。（依照字母表顺序）

sort -k 3                    基于第 3 字段排序  
sort -u                    排序的同时去除重复行。[unique]

##### 4) cut

提取文本文件中字段

cut -d ' ' -f 3-4 文本文件名    提取文件中以空格符分割的第 3-4 个字段  
cut -f 2 文本文件名            提取文件中第 2 个字段

<code>cut -f 2-4 文本文件名</code>	提取文件中第 2-4 个字段
<code>cut -f 2,4 文本文件名</code>	提取文件中第 2 和第 4 个字段
<code>cut -c 3-6 文本文件名</code>	提取文件中第 3-6 个字符

#### 4) **paste**

合并文本文件

<code>paste 文件 1 文件 2 &gt;文件 3</code>	逐行合并文件 1 和文件 2 并创建文件 3
<code>cat 文件名   paste - -</code>	将文件中每两行合并到一行显示

#### 5) **wc**

统计并显示文本文件行数，由空格分隔的单词数和字符数

`wc -l`      显示行数

#### 6) **grep**

globally search a regular expression and print

<code>grep "字符串" 文件名</code>	检索并显示文件中包含字符串的行
<code>grep "^字符串" 文件名</code>	检索并显示文件中以字符串开头的行
<code>grep "seq\$"</code>	检索并显示含以 seq 结尾的行

#### 7) **top**

动态显示系统运行状态，包括处理器、内存使用情况以及进程，q 退出

#### 8) **ps**

动态显示系统运行状态，包括处理器、内存使用情况以及进程，q 退出

`ps -u`      显示本用户当前进程信息

```

le4b@bbt:~$ ps -u
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
le4b      11109  0.1  0.0  22680  5036 pts/2    Ss   15:50   0:00 -bash
le4b      11125  0.0  0.0  37364  3316 pts/2    R+   15:50   0:00 ps -u
le4b@bbt:~$

```

#### 9) **kill**

`kill 31352`      终止编号为 31352 的无用进程

#### 10) **history**

显示最近用过的命令

#### 11) **alias**

用自定义字符串来取代命令

`alias ll="ls -l"`      用别名 ll 取代命令 `ls -l`

#### 12) **man**

查看命令帮助手册

## 6.2 文本编辑器 nano 使用的复习

```
GNU nano 2.9.3 New Buffer
[Green cursor]

^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos  M-U Undo
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line M-E Redo
```

在绿色光标处输入文本

^O: 编辑文件名, 选择保存格式并保存

```
File Name to Write: file
^G Get Help  M-D DOS Format  M-A Append  M-B Backup File
^C Cancel    M-M Mac Format  M-P Prepend  ^T To Files
```

^R: 读入文件

^W: 在文件中搜索内容

^K: 剪切一整行

^U: 粘贴文本

M-U: 撤销

M-E: 重做

## 6.3 EMBOSS 中序列比对工具 needle 的使用复习

```
leb4b@bbt:~/0327/data$ needle
Needleman-Wunsch global alignment of two sequences
Input sequence: HBA_HUMAN.FASTA
Second sequence(s): HBA_MOUSE.FASTA
Gap opening penalty [10.0]:
Gap extension penalty [0.5]:
Output alignment [hba_human.needle]: HUMAN_MOUSE.NEEDLE
```

输入 needle 打开序列比对界面

分别输入两条序列所在的文本文件名

设置空位起始和空位延伸罚分值, 不输入则为默认值 10 和 0.5

设有连续的  $n$  个空位:  $\text{gap opening penalty} + (n - 1) * \text{gap extension penalty}$   
设置比对结果输出的文件名

序列比对结果:

```
#=====
#
# Aligned_sequences: 2
# 1: HBA_HUMAN
# 2: HBA_MOUSE
# Matrix: EBL0SUM62
# Gap_penalty: 10.0
# Extend_penalty: 0.5
#
# Length: 142
# Identity:      122/142 (85.9%)
# Similarity:    131/142 (92.3%)
# Gaps:          0/142 ( 0.0%)
# Score: 648.0
#
#
#=====

HBA_HUMAN      1  MVLSPADKTNVKAAWGKVGAGHAGEYGAEALERMFSLFPTTKTYFPHFDLS      50
      |||.:.||:|:|||||:|.:.|||||||.|.|||||||:|
HBA_MOUSE      1  MVLSGEDKSNIAAWGKIGGHGAEYGAEALERMFASFPTTKTYFPHFDVS      50

HBA_HUMAN     51  HGSAQVKGHGKKVADALTNAVAHVDDMPNALSALSDDLHAHKLRVDPVNFK      100
      |||||:|.:.||:|:|.|||||||.|.|||||||
HBA_MOUSE     51  HGSAQVKGHGKKVADALASAAGHLDDLPGALSALSDDLHAHKLRVDPVNFK      100

HBA_HUMAN    101  LLSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTSKYR      142
      |||||:|.:.||:|:|||||||.|.|||||||
HBA_MOUSE    101  LLSHCLLVTLASHHPADFTPAVHASLDKFLASVSTVLTSKYR      142

#-----
#-----
```

Identity: 一致性, 两条序列完全相同的氨基酸比例

Similarity: 相似性, 两条序列具有性质相似氨基酸的比例

## 7. 收获

通过本次讨论, 我们复习了上节课学习的 Linux 指令操作以及 nano 文本编辑器的使用方法。此外, 初步认识了序列比对。

## 8. 问题: 无

## 9. 建议: 无