演習問題　解答編

復習問題１　UNIX 基本コマンド

1.
アプリケーション ＞ ユーティリティ ＞ ターミナル

```
$ ssh bias4.nibb.ac.jp
$ cd data/1_unix
$ pwd
$ ls -R
```

2. 以下は、先に~/unixtest ディレクトリに移動してから操作する場合となります。

```
$ cd ~/unixtest
$ mkdir FASTA-EX
$ cp ~/data/1_unix/sprot/*.fasta FASTA-EX
$ ls ~/unixtest/FASTA-EX
```

3. 以下は、~/unixtest がカレントディレクトリの場合です。

```
$ cd FASTA-EX
$ man grep  、"/"で検索
$ grep -h '^>' * | less
```

4. tail コマンドの引数に "-n +行数" を指定すると、先頭から数えて、指定された行数以降を表示します。

```
$ man tail  、"/"で検索
$ tail -n +2 -q * | less
```

5. カレントディレクトリに注意して、結果ファイルがどこに作成されるかをよく考えて実行します。

```
$ cat *HUMAN.fasta > ../human.fasta
$ cat *MOUSE.fasta > ../mouse.fasta
$ cat *DROME.fasta > ../drome.fasta
```

6. 以下は、~/unixtest/FASTA-EX がカレントディレクトリの場合です。

```
$ cd ..
$ tar zcvf FASTA-EX.tar.gz ./FASTA-EX
$ tar ztvf FASTA-EX.tar.gz
$ rm -rf FASTA-EX
```

上記のようにした場合は、解凍した際に FASTA-EX ディレクトリが作成されてその下にファイル群が配置される。一方

```
$ tar zcvf ../FASTA-EX.tar.gz .
```

のように対象ディレクトリにカレントディレクトリ(.)を指定すると、解凍時にカレントディレクトリに直に大量のファイルが作成されてしまうので注意すること。

7. chmod u+rw,g-rwx,o-rwx FASTA-EX.tar.gz でもよい。

```
$ chmod 600 FASTA-EX.tar.gz
```

*8. find コマンドは検索の起点になるディレクトリパスを最初に指定し、評価式 -name の後ろにファイル名を指定することによって評価式に従ってディレクトリツリーを検索します。この時指定するファイル名にはワイルドカード等を使用したパターンマッチによる検索もできます（9.で使用)。

```
$ find ~ -name SYYM_DROME.sprot
  /home/miwa/data/1_unix/sprot/ext/flybase/SYYM_DROME.sprot
```

*9. -exec アクションは、それ以降に記述されたコマンドから";"までをコマンドラインとして実行でき、検索結果のファイルをコマンドの引数として使用することができます。'{}'は検索結果の中で現在の処理対象となるファイルに置き換えられます。ある条件に合致するディレクトリツリー下にあるファイル全てに対して特定の処理を行いたい時に便利です。

```
$ mkdir ~/SP
$ find ~ -name '*.sprot' -exec cp {} ~/SP ¥;
```

復習問題２　エディタとスクリプト

１）（テキストを参照せよ）

２）（略）

３）実行結果は以下のようになる。

```
GO matchs;
4 1433T_MOUSE.sprot.tmp
```

実行できない場合、以下のようにして実行権を付与すること。

```
$ chmod +x example4.sh
```

４） 以下に一例を挙げる。
この例では最後に .tmp ファイルを rm コマンドで消去するようにしている。

```
#!/bin/sh
param=$1
grep ${param} 1433T_HUMAN.sprot > line.tmp
echo "${param} matchs;"
wc -l line.tmp
rm line.tmp
```

復習問題３　UNIX によるテキストファイル処理

1)　$ awk '$2>=100{print}' ecoli.htseq

2)　$ awk '$2>=100{print}' ecoli.htseq | grep ‘^b’ > ecoli.temp

3)　$ sort –k 2,2nr ecoli.temp > ecoli.htseq.sorted

4)

```
awk '$2>=100{print}' ecoli.htseq | grep ‘^b’ > ecoli.temp
sort –k 2,2nr ecoli.temp > ecoli.htseq.sorted
rm ecoli.temp
```

最終行に rm ecoli.temp を加える事により、中間ファイルを削除できる。

別解

awk '$2>=100{print}' ecoli.htseq | grep ‘^b’ | sort –k 2,2nr > ecoli.htseq.sorted

5)

```
filename=$1
awk ‘$2>=100{print}’ $filename | grep ‘^b’ > ecoli.temp
sort –k 2,2nr ecoli.temp
rm ecoli.temp
```

$ ./htsort.sh ecoli_other.htseq を実行して確認すること。

復習問題４　SunGridEngine 利用方法＆シェルスクリプト

1. スクリプト内容（ans1.sh）

```
#!/bin/sh
#$ -t 1-12
#$ -cwd

bowtie2 -x ../../4_sge/genome/ecoli_genome
     -U eco_${SGE_TASK_ID}.fq -S results/eco_${SGE_TASK_ID}.sam
```

実行： `qsub ans1.sh`

2. 確認用：スクリプト内容（ans2.sh）

```
#!/bin/sh
i=1
for org in *.fastq
  do
    newname=${org/.fastq/.sam}
    echo "mv results/eco_${i}.sam results/${newname}"
    i=$((i+1))
done
```

実行用：スクリプト内容

```
#!/bin/sh
i=1
for org in *.fastq
  do
    newname=${org/.fastq/.sam}
    mv results/eco_${i}.sam results/${newname}
    i=$((i+1))
done
```

このスクリプトは、

1. 変数「i」に 1 を代入する
2. 「元のファイル名（`CFT_MP-1.fastq`〜`MG_Ur-3.fastq`）」を for 文のリストとし、一つづつ変数 org に代入される
3. シェルの変数展開を用いて、拡張子が「.sam」のファイル名を生成する
4. mv コマンドで、eco_${i}.sam を CFT_MP-1.sam に変更
5. 以下、変数 i をインクリメントしながら繰り返し

for のリストを生成する際、extra.sh でシンボリックリンクを作った時と同じことをしているので、「必ず同じ順番で」リストになり echo_1〜echo_12 の番号と対応させることができます。

実践演習１　RNA-Seq 解析結果の集計

1) ファイルが 1,2,3,...の順に表示されるのは B)の方。

```
$ paste results/ecoli.{?,1?}.htseq > ecoli.paste_all
```

2) 遺伝子にヒットしなかったリードの情報の行は、_で始まっているのでこれを除く。

grep –v はマッチする行を除いて出力する。

```
$ cut –f 1,2,4,6,8,10,12,14,16,18,20,22,24 ecoli.paste_all | grep -v
'^_' > ecoli.count_all
```

3) 長さは （終了位置）—（開始位置）＋ 1 で計算される

```
$ awk '$3=="exon" {print $10, $5-$4+1}' ecoli.gtf | sed 's/[";]//g' >
ecoli.gene_length
```

4) 行の先頭からアルファベット順に並べたい場合は sort のオプションは不要。また、join のキーがどちらも 1 列目であるときは join のオプションは不要。

```
$ sort ecoli.gene_length > ecoli.gene_length_sorted
$ join ecoli.gene_length_sorted  ecoli.count_all | sed 's/ /¥t/g' >
ecoli.result_all
```

5) 最初に遺伝子名（1 列目）を出力、それ以降は 3,4,5 列目、6,7,8 列目、9,10,11 列目、12,13,14 列目のそれぞれ平均をとり、遺伝子の長さ（2 列目）で割って 1000 倍する。

```
$ awk '{ print $1, ($3+$4+$5)/3/$2*1000, ($6+$7+$8)/3/$2*1000,
($9+$10+$11)/3/$2*1000,  ($12+$13+$14)/3/$2*1000  }'  ecoli.result_all
>ecoli.result_mean
```

実践演習２ BLAST を用いた比較ゲノム解析

2-1) E-value は 11 カラム目。

```
$ awk '$11 < 1e-20 {print}' eco_sal.blast >eco_sal.cut20.blast
```

2-2) データベース配列の開始位置、終了位置はそれぞれ 9 カラム目と１０カラム目。

```
$ awk '$9 > $10 {print}' eco_sal.blast |wc
```

330 個ある

2-3) awk と sort をパイプで組み合わせる。大腸菌はクエリなので 7 、 8 カラム目を見る

```
$ awk '$7 <= 1280000 && $8 >= 1260000 {print}' eco_sal.blast | sort –k 7,7n
```

注）一見 `awk '$7 >= 1260000 && $8 <= 1280000 {print}'`でもいいように見えるが、これだと問題文の図で B の場合しか満たさない。