

Advanced Linux Usage

160920 Martin Dahlö martin.dahlo@scilifelab.uu.se

Enabler for Life Science











```
$ ls -l
total 0
-rw-rw-r-- 1 dahlo dahlo 0 Sep 1 16:42 sample_1.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep 1 16:42 sample_2.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep 1 16:42 sample_3.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep 1 16:42 sample_4.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep 1 16:42 sample_5.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep 1 16:42 sample_6.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep 1 16:42 sample_7.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep 1 16:42 sample_8.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep 1 16:42 sample_9.bam
```



```
$ ls -l
total 0
-rw-rw-r-- 1 dahlo dahlo 0 Sep    1 16:42 sample_1.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep    1 16:42 sample_2.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep    1 16:42 sample_3.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep    1 16:42 sample_4.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep    1 16:42 sample_5.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep    1 16:42 sample_6.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep    1 16:42 sample_7.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep    1 16:42 sample_8.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep    1 16:42 sample_9.bam
$ my_prog sample_1.bam
```



```
$ ls -l
total 0
-rw-rw-r-- 1 dahlo dahlo 0 Sep    1 16:42 sample_1.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep    1 16:42 sample_2.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep    1 16:42 sample_3.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep    1 16:42 sample_4.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep    1 16:42 sample_5.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep    1 16:42 sample_6.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep    1 16:42 sample_7.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep    1 16:42 sample_8.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep    1 16:42 sample_9.bam
$ my_prog sample_1.bam
$ my_prog sample_2.bam
```



```
s ls -l
total 0
-rw-rw-r-- 1 dahlo dahlo 0 Sep 1 16:42 sample 1.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep 1 16:42 sample 2.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep 1 16:42 sample 3.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep 1 16:42 sample 4.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep 1 16:42 sample 5.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep 1 16:42 sample 6.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep 1 16:42 sample 7.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep 1 16:42 sample 8.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep 1 16:42 sample 9.bam
$ my prog sample 1.bam
$ my prog sample 2.bam
$ my prog sample 3.bam
$ my prog sample 4.bam
$ my prog sample 5.bam
$ my prog sample 6.bam
$ my prog sample 7.bam
$ my prog sample 8.bam
$ my prog sample 9.bam
```



- Same program, many files
 - 10 files? Ok
 - 1000 files? Not ok..



- Same program, many files
 - 10 files? Ok
 - 1000 files? Not ok..
- Reproducibility
 - Self and others



- Same program, many files
 - 10 files? Ok
 - 1000 files? Not ok..
- Reproducibility
 - Self and others
- As always, scripts to the rescue!



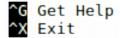
```
total 0
-rw-rw-r-- 1 dahlo dahlo 0 Sep 1 17:18 sample_1.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep 1 17:18 sample_2.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep 1 17:18 sample_3.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep 1 17:18 sample_4.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep 1 17:18 sample_5.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep 1 17:18 sample_6.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep 1 17:18 sample_7.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep 1 17:18 sample_8.bam
-rw-rw-r-- 1 dahlo dahlo 0 Sep 1 17:18 sample_9.bam
s nano analysis.sh
```

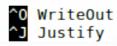


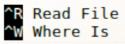
GNU nano 2.0.9

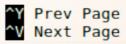
File: analysis.sh

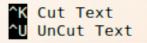
Modified

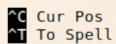














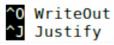
GNU nano 2.0.9

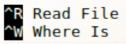
File: analysis.sh

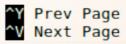
Modifie

my_prog sample_1.bam

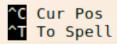
^G Get Help ^X Exit











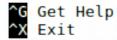


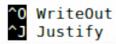
GNU nano 2.0.9

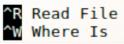
File: analysis.sh

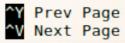
Modifie

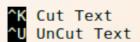
my_prog sample_1.bam
my_prog sample_2.bam

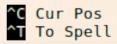








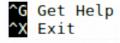


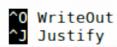


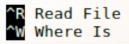


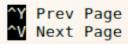
GNU nano 2.0.9 File: analysis.sh Modified

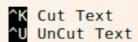
my_prog sample_1.bam my_prog sample_2.bam my_prog sample_3.bam my_prog sample_4.bam my_prog sample_5.bam my_prog sample_6.bam my_prog sample_7.bam my_prog sample_8.bam my_prog sample_9.bam

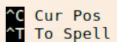














```
s l
total 4,0K
-rw-rw-r-- 1 dahlo dahlo 267 Sep 7 09:34 analysis.sh
-rw-rw-r-- 1 dahlo dahlo
                           0 Sep 1 17:18 sample 1.bam
-rw-rw-r-- 1 dahlo dahlo
                           0 Sep 1 17:18 sample 2.bam
                           0 Sep 1 17:18 sample 3.bam
-rw-rw-r-- 1 dahlo dahlo
-rw-rw-r-- 1 dahlo dahlo
                           0 Sep 1 17:18 sample 4.bam
-rw-rw-r-- 1 dahlo dahlo
                           0 Sep 1 17:18 sample 5.bam
-rw-rw-r-- 1 dahlo dahlo
                           0 Sep 1 17:18 sample 6.bam
-rw-rw-r-- 1 dahlo dahlo
                           0 Sep 1 17:18 sample 7.bam
-rw-rw-r-- 1 dahlo dahlo
                           0 Sep 1 17:18 sample 8.bam
-rw-rw-r-- 1 dahlo dahlo
                           0 Sep 1 17:18 sample 9.bam
```



```
s l
total 4,0K
-rw-rw-r-- 1 dahlo dahlo 267 Sep 7 09:34 analysis.sh
-rw-rw-r-- 1 dahlo dahlo
                           0 Sep 1 17:18 sample 1.bam
-rw-rw-r-- 1 dahlo dahlo
                           0 Sep 1 17:18 sample 2.bam
-rw-rw-r-- 1 dahlo dahlo
                           0 Sep 1 17:18 sample 3.bam
-rw-rw-r-- 1 dahlo dahlo
                           0 Sep 1 17:18 sample 4.bam
-rw-rw-r-- 1 dahlo dahlo
                           0 Sep 1 17:18 sample 5.bam
-rw-rw-r-- 1 dahlo dahlo
                           0 Sep 1 17:18 sample 6.bam
-rw-rw-r-- 1 dahlo dahlo
                           0 Sep 1 17:18 sample 7.bam
-rw-rw-r-- 1 dahlo dahlo
                           0 Sep 1 17:18 sample 8.bam
-rw-rw-r-- 1 dahlo dahlo
                           0 Sep 1 17:18 sample 9.bam
$ bash analysis.sh
```



Assigning

```
my_variable=5
my variable="nice text"
```



Assigning

```
my_variable=5
my variable="nice text"
```

Using

```
$my_variable
```



Assigning

```
my_variable=5
my variable="nice text"
```

Using

```
$my variable
```

\$ my_variable="Martin"



Assigning

```
my_variable=5
my variable="nice text"
```

Using

```
$my variable
```

```
$ my_variable="Martin"
$ echo "Hello $my_variable"
```



Assigning

```
my_variable=5
my_variable="nice text"
```

Using

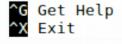
```
$my variable
```

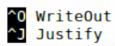
```
$ my_variable="Martin"
$ echo "Hello $my_variable"
Hello Martin
```

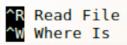


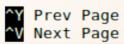
GNU nano 2.0.9 File: analysis.sh Modified

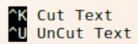
my_prog sample_1.bam my_prog sample_2.bam my_prog sample_3.bam my_prog sample_4.bam my_prog sample_5.bam my_prog sample_6.bam my_prog sample_7.bam my_prog sample_8.bam my_prog sample_9.bam

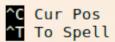










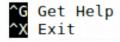


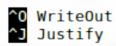


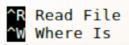
GNU nano 2.0.9 File: analysis.sh Modified

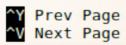
prefix="sample"

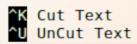
my_prog sample_1.bam my_prog sample_2.bam my_prog sample_3.bam my_prog sample_4.bam my_prog sample_5.bam my_prog sample_6.bam my_prog sample_7.bam my_prog sample_8.bam my_prog sample_9.bam

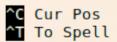










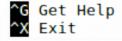


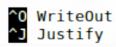


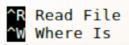
GNU nano 2.0.9 File: analysis.sh Modified

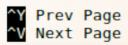
prefix="sample"

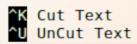
my_prog \${prefix}_1.bam
my_prog \${prefix}_2.bam
my_prog \${prefix}_3.bam
my_prog \${prefix}_4.bam
my_prog \${prefix}_5.bam
my_prog \${prefix}_6.bam
my_prog \${prefix}_7.bam
my_prog \${prefix}_7.bam
my_prog \${prefix}_8.bam
my_prog \${prefix}_9.bam

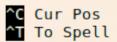










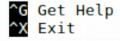


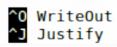


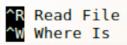
GNU nano 2.0.9 File: analysis.sh Modified

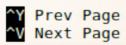
prefix="dog"

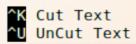
my_prog \${prefix}_1.bam
my_prog \${prefix}_2.bam
my_prog \${prefix}_3.bam
my_prog \${prefix}_4.bam
my_prog \${prefix}_5.bam
my_prog \${prefix}_6.bam
my_prog \${prefix}_7.bam
my_prog \${prefix}_7.bam
my_prog \${prefix}_8.bam
my_prog \${prefix}_9.bam

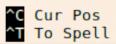














```
for variable_name in 1 2 3;
do
    echo $variable_name
done
```



```
for variable_name in text works too;
do
```

echo \$variable_name done

```
$ bash loop_test.sh
text
works
too
$
```



```
for variable_name in mix them 5;
do
    echo $variable_name
done

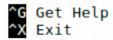
$ bash loop_test.sh
mix
them
5
```

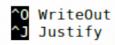


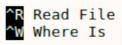
GNU nano 2.0.9

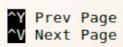
File: analysis.sh

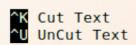
prefix="sample"
for i in 1 2 3 4 5 6 7 8 9;
do
 my_prog \${prefix}_\$i.bam
done

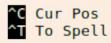












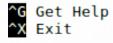


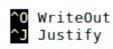
GNU nano 2.0.9

File: analysis.sh

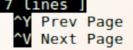
prefix="sample"

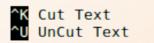
for i in 1 2 3 4 5 6 7 8 9;
 do
 echo my_prog \${prefix}_\$i.bam
 done

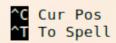














GNU nano 2.0.9

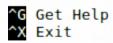
File:

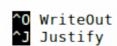
prefix="sample"

for i in 1 2 3 4 5 6 7 8 9;
 do
 echo my_prog \${prefix}_\$i.bam
 done

```
Loops
```

```
$ bash analysis.sh
my_prog sample_1.bam
my_prog sample_2.bam
my_prog sample_3.bam
my_prog sample_4.bam
my_prog sample_5.bam
my_prog sample_6.bam
my_prog sample_7.bam
my_prog sample_8.bam
my_prog sample_9.bam
$
```









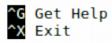
```
$ ls *.bam
sample_1.bam sample_3.bam sample_5.bam sample_7.bam sample_9.bam
sample_2.bam sample_4.bam sample_6.bam sample_8.bam
```

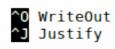


GNU nano 2.0.9

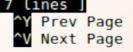
File: analysis.sh

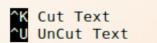
```
prefix="sample"
for file in $( ls *.bam );
do
    echo my_prog $file
done
```

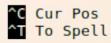










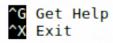


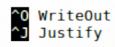


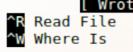
GNU nano 2.0.9

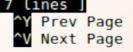
File: analysis.sh

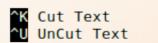
```
for file in $( ls *.bam );
do
    echo my_prog $file
done
```

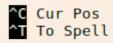














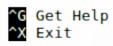
GNU nano 2.0.9

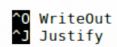
File:

```
for file in $( ls *.bam );
do
    echo my_prog $file
done
```

Loop over files

```
$ bash analysis.sh
my_prog sample_1.bam
my_prog sample_2.bam
my_prog sample_3.bam
my_prog sample_4.bam
my_prog sample_5.bam
my_prog sample_6.bam
my_prog sample_7.bam
my_prog sample_8.bam
my_prog sample_9.bam
$
```









\$ bash analysis.sh /path/to/my/bams



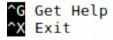
Loop over files

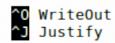
GNU nano 2.0.9

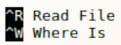
File: analysis.sh

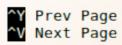
Modified

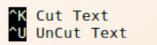
```
for file in $( ls $1/*.bam );
do
    echo my_prog $file
done
```

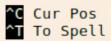














GNU nano 2.0.9

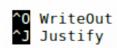
File:

```
for file in $( ls $1/*.bam );
do
    echo my_prog $file
done
```

Loop over files

```
$ bash analysis.sh .
my_prog ./sample_1.bam
my_prog ./sample_2.bam
my_prog ./sample_3.bam
my_prog ./sample_4.bam
my_prog ./sample_5.bam
my_prog ./sample_6.bam
my_prog ./sample_7.bam
my_prog ./sample_8.bam
my_prog ./sample_9.bam
$
```

```
^G Get Help
^X Exit
```







GNU nano 2.0.9

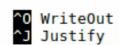
File:

```
for file in $( ls $1/*.bam );
do
    echo my_prog $file
done
```

Loop over files

```
$ bash analysis.sh /path/to/my/bams
my_prog /path/to/my/bams/sample_1.bam
my_prog /path/to/my/bams/sample_2.bam
my_prog /path/to/my/bams/sample_3.bam
my_prog /path/to/my/bams/sample_4.bam
my_prog /path/to/my/bams/sample_5.bam
my_prog /path/to/my/bams/sample_6.bam
my_prog /path/to/my/bams/sample_7.bam
my_prog /path/to/my/bams/sample_8.bam
my_prog /path/to/my/bams/sample_9.bam
s
```

```
^G Get Help
^X Exit
```



```
^R Read F
^W Where
```



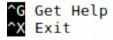
Loop over files

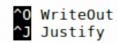
GNU nano 2.0.9

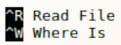
File: analysis.sh

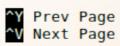
Modified

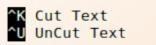
```
for file in $( ls $1/*.bam );
do
    my_prog $file
done
```

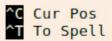














Loop over files

```
$ my_prog sample_1.bam
$ my_prog sample_2.bam
$ my_prog sample_3.bam
$ my_prog sample_4.bam
$ my_prog sample_5.bam
$ my_prog sample_6.bam
$ my_prog sample_7.bam
$ my_prog sample_8.bam
$ my_prog sample_9.bam
```

```
for file in $( ls $1/*.bam );
do
    my_prog $file
done
```

```
if true; then
  echo "This is true"
fi
```

```
if false; then
  echo "This is true"
fi
```

```
if [[ 5 < 9 ]]; then
  echo "This is true"
fi</pre>
```

```
if [[ 5 > 9 ]]; then
  echo "This is true"
fi
```

```
if [[ 5 == 9 ]]; then
  echo "This is true"
fi
```

```
if [[ "Hello" == "Hello" ]]; then
  echo "This is true"
fi
```

```
if [[ "Hello" == "Hi" ]]; then
  echo "This is true"
fi
```

```
if [[ "Hello" == "Hel"* ]]; then
  echo "This is true"
fi
```



```
for file in $( ls $1/*.bam );
do
    echo my_prog $file
done
```



```
for file in $( ls $1/*.bam );
do
    if [[ ... != "dog"* ]]; then
       echo my_prog $file
    fi
done
```

```
for file in $( ls $1/*.bam );
do
    if [[ ... != "dog"* ]]; then
       echo my_prog $file
    fi
done
```

Ex: \$file is /path/to/dog_1.bam

```
for file in $( ls $1/*.bam );
do
    if [[ ... != "dog"* ]]; then
       echo my_prog $file
    fi
done
```

Ex: \$file is /path/to/dog_1.bam basename \$file

```
for file in $( ls $1/*.bam );
do
    if [[ ... != "dog"* ]]; then
       echo my_prog $file
    fi
done
```

Ex: \$file is /path/to/dog_1.bam basename \$file dog 1.bam

```
for file in $( ls $1/*.bam );
do
    if [[ $(basename $file) != "dog"* ]]; then
        echo my_prog $file
    fi
done
```

Ex: \$file is /path/to/dog_1.bam basename \$file dog 1.bam

```
for file in $( ls $1/*.bam );
do
    if [[ $(basename $file) != "dog"* ]]; then
        my_prog $file
    fi
done
```

Ex: \$file is /path/to/dog_1.bam basename \$file dog_1.bam



- Programming is programming
 - Perl, Python, Bash, and more



- Programming is programming
 - Perl, Python, Bash, and more

```
for file in $( ls $1/*.bam );
do
    if [[ $(basename $file) != "dog"* ]]; then
        my_prog $file
    fi
done
```



- Programming is programming
 - Perl, Python, Bash, and more



- Programming is programming
 - Perl, Python, Bash, and more



- Programming is programming
 - Perl, Python, Bash, and more
- Start with one, git gud, (learn another)



- Programming is programming
 - Perl, Python, Bash, and more
- Start with one, git gud, (learn another)

PYTHON



Laboratory time! (yet again)