

Bash Script Examples

Learn how to use powerful Bash scripts to enhance your command line experience. In this presentation, we will explore two scripts: one for displaying lines within a range and another for deleting lines with a specific word.

```
Processes are controlled by a file RUN_STEPS

xt

g level set for $DEBUG_LEVEL" >> ${LOGRUN_SOM

=run_all_somatic_SNV_steps

-s ${RUN_STEPS}

compute_low_somatic_snp" >${RUN_STEPS}
compute_high_somatic_snp" >>${RUN_STEPS}
erge_low_N_high" >>${RUN_STEPS}
opy_merged_file" >>${RUN_STEPS}
nnotate_putative_mutation" >>${RUN_STEPS}
n_gene_annotation" >>${RUN_STEPS}
reate_excel_sheet" >>${RUN_STEPS}
nnotate_low_tier_mutation" >>${RUN_STEPS}
odify_SJLLQ" >>${RUN_STEPS}

JG: created run_all_somatic_SNV_steps" >> $LOG

JG_LEVEL -gt 0 ]

JG: created run_all_somatic_SNV_steps" >> $LOG
```

Script 1: Displaying Lines

Let's start with a script that allows you to display lines within a specified range in a file. This can be helpful when you want to focus on specific content without the distractions of the entire file.

Here's the Bash code for the script:

```
echo "enter file name: "  
read fname  
echo "enter the starting line number: "  
read s  
echo "enter the ending line number: "  
read e  
# sed -n $s,$e\p $fname  
awk "NR >= $s && NR <= $e" "$fname"
```

Example for Script 1

Let's consider an example to understand how Script 1 works. Assume we have a file named 'example.txt' with the following content:

```
text file: Line 1 Line 2 Line 3 Line 4 Line 5
```

Script 2: Deleting Lines with a Word

Now, let's explore a script that allows you to delete lines containing a specific word in one or multiple files. This can be useful when you want to remove unnecessary information or sensitive data from your files.

Here's the Bash code for the script:

```
echo enter number of files
read n
echo enter word
read word
for ((i = 1; i <= $n; i++)); do
    echo enter file name
    read file
    cat $file
    sed -i "/$word/d" "$file"
    cat $file
done
```