Workshop 1: **Introduction to UNIX command-line**

Day 3

Serghei Mangul, PhD | smangul@ucla.edu

CBI Fellow

“Swiss Army knife” set of tools

Summary of Day 2

file permissions

cat

wc

>, >>, <

pipeline

ln –s

grep

regex

sed

awk

- awk is both a
  - programming language
  - text processor

whitespace (spaces, tabs, etc.) to separate fields

- parsing and manipulating **tabular** data
  - iterates through the entire file line-by-line

```
awk '{action_to_take}' <file_to_parse>
```

awk : Simple Uses

action_to_take

```
awk '{print}’ <file_to_parse>
```

```
awk '{print $1}’ <file_to_parse>
```

column number

```
awk '{print $1”\t”$3}’ <file_to_parse>
```

delimiter

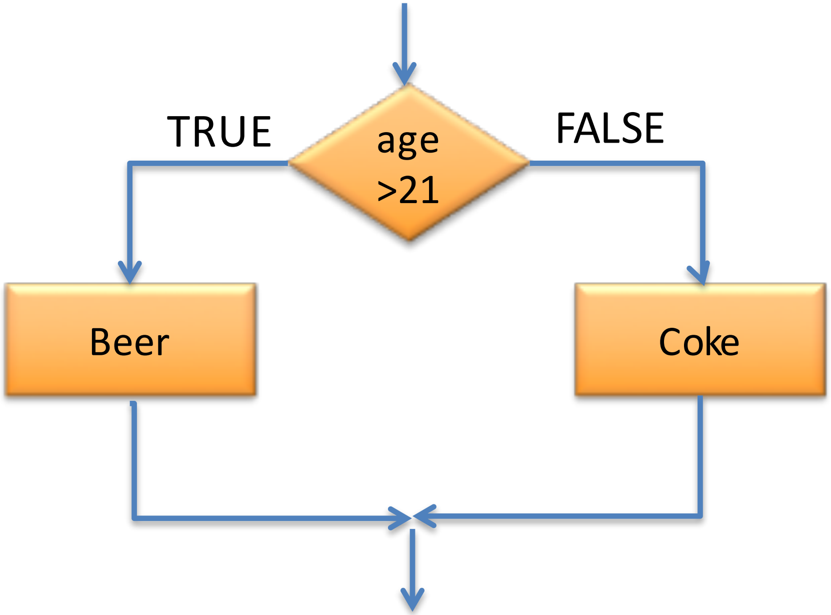
Let’s practice

```
awk '{print}' hg19.gtf
awk '{print $1}’ hg19.gtf
awk '{print $4}' hg19.gtf
awk '{print $1"\t"$4} hg19.gtf
awk '{print $1"\t"$4*$5}' hg19.gtf
```

awk : If statement

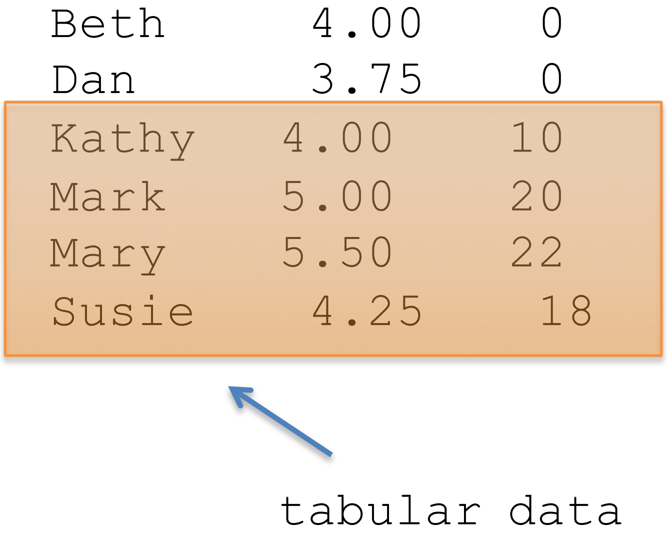
condition

```
if(age>21) then
    print("Beer Please")
```



awk : If statement

Print those employees who actually worked



```
Beth      4.00    0
Dan       3.75    0
Kathy    4.00    10
Mark     5.00    20
Mary     5.50    22
Susie     4.25    18
```

tabular data

```
awk ‘{if($3>0) print}’ emp.data
```

Data type

- Numbers
- **Text**

```
1        clothing      3141
1        computers     9161
1        textbooks     21312
2        clothing      3252
2        computers     12321
2        supplies      2242
2        textbooks     15462
```

Print information about computers only

```
awk '{if($2=="computers") print}' sales.dat
```

use double quote

Text

Let’s practice!

```
awk '{if($3>0) print}' emp.data
awk '{if($3>0) print $1"\t"$2*$3}' emp.data
```

```
Beth     4.00    0
Dan      3.75    0
Kathy   4.00    10
Mark    5.00    20
Mary    5.50    22
Susie    4.25    18
```

Let’s practice!

```
awk '{if($4>50) print}’ hg19.gtf
awk '{if($1=="chr2") print}’ hg19.gtf
awk '{if($1=="chr2") print $1"\t"$3}’ hg19.gtf
```

```
chr2	hg18_knownGene_GnfAtlas2	exon	237538	237602	0.000000	-	.	gene_id "204019_s_at"; transcript_id "uc002qvu.1";
chr2	hg18_knownGene_GnfAtlas2	exon	239731	239852	0.000000	-	.	gene_id "204019_s_at"; transcript_id "uc002qvu.1";
```

awk : sum

- Calculate sum of a particular column

```
awk '{sum+=$2} END { print sum}' emp.data
```

New variable

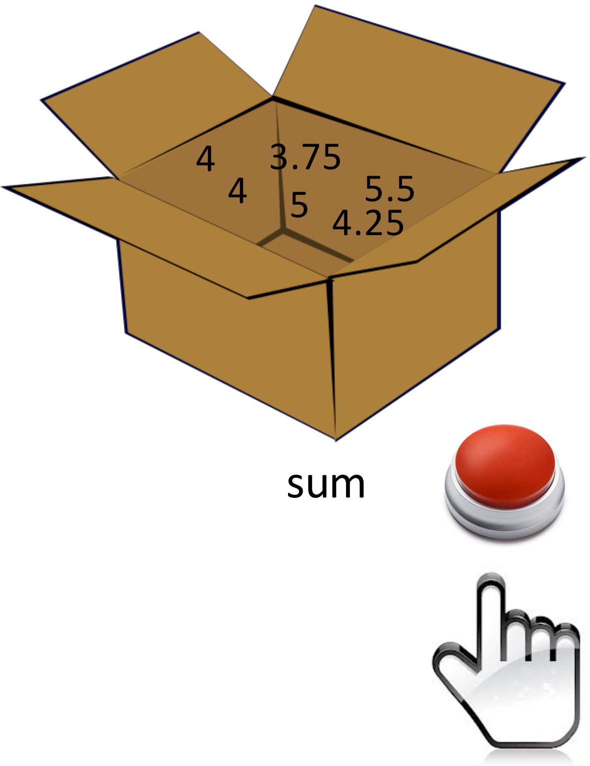
“+=” means to add the value to the variable sum

Do it until the end of the file

A variable is like a **box** where we can store a value and reuse this same value multiple times in our program.

Variable is like a box

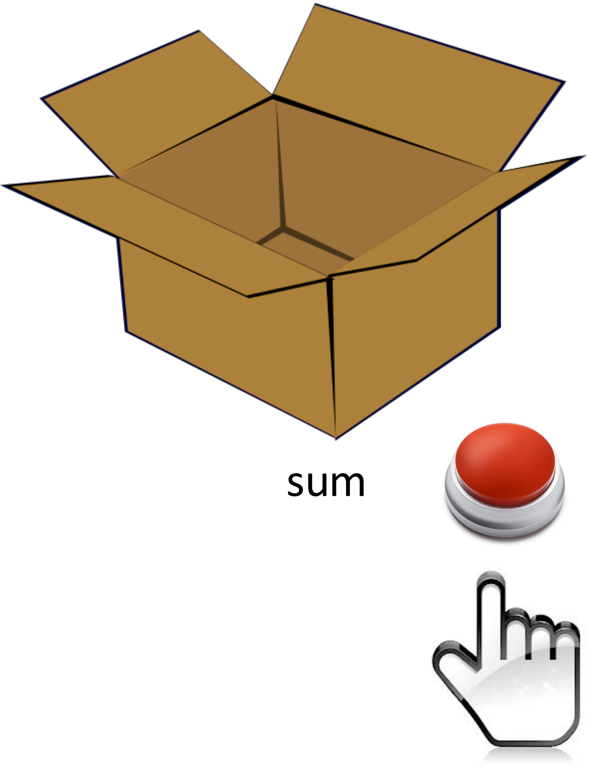
```
awk '{sum+=$2} END { print sum}' emp.data
```



```
Beth     4.00    0
Dan      3.75    0
Kathy   4.00    10
Mark    5.00    20
Mary    5.50    22
Susie    4.25    18
```

Variable is like a box

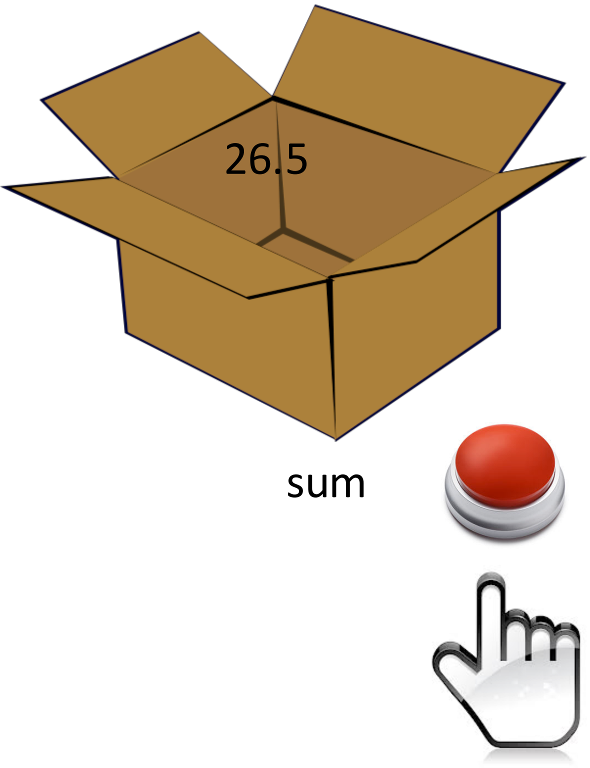
```
awk '{sum+=$2} END { print sum}' emp.data
```



```
Beth     4.00    0
Dan      3.75    0
Kathy   4.00    10
Mark    5.00    20
Mary    5.50    22
Susie    4.25    18
```

Variable is like a box

```
awk '{sum+=$2} END { print sum}' emp.data
```



```
Beth      4.00    0
Dan       3.75    0
Kathy    4.00    10
Mark     5.00    20
Mary     5.50    22
Susie     4.25    18
```

Let’s practice!

```
awk '{sum+=$4} END {print sum}’ hg19.gtf

awk ' END {print NR}' hg19.gtf
wc –l hg19.gtf

awk '{sum+=$4} END {print sum/3000}’ hg19.gtf
awk '{sum+=$4} END {print sum/NR}’ hg19.gtf
awk '{if($1=="chr2") sum+=$4} END {print sum}’ hg19.gtf
```

```
chr2    hg18_knownGene_GnfAtlas2    exon    237538  237602  0.000000    -   .   gene_id "204019_s_at"; transcript_id "uc002qvu.1";
chr2    hg18_knownGene_GnfAtlas2    exon    239731  239852  0.000000    -   .   gene_id "204019_s_at"; transcript_id "uc002qvu.1";
```

Sort

- will rearrange the lines in a text file so that they are sorted, numerically and alphabetically.

```
sort [OPTION]... [FILE]...
```

Options :

- -n - numerical ordering
- -k - sort by a particular column

Sort a file

regex.txt

```
beat
brat
boat
bat
banana
```

```
sort regex.txt
```

Sort a text file

```
sort -n regex.txt
```

Sort a file with numbers

Sort by column

```
sort -k 2 sales.dat
```

Column number

```
sort -k 3n sales.dat
```

sales.dat

```
1     clothing     3141
1     computers     9161
1     textbooks     21312
2   clothing     3252
3   …
```

Uniq

- Removes duplicate lines from a file*

```
uniq [OPTION]... [INPUT]
```

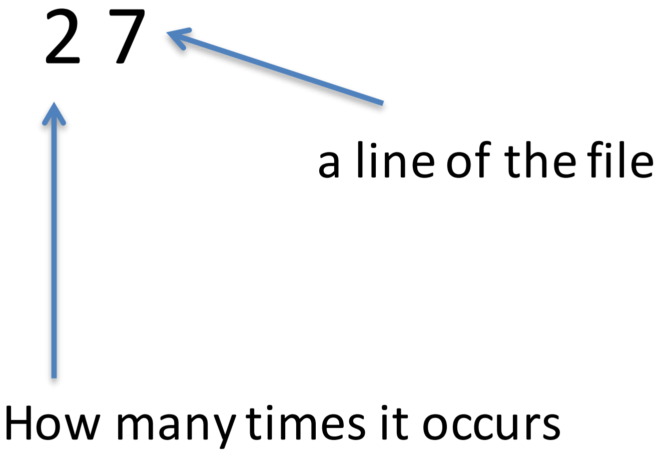
Options :

- -c - how many times each line occurred
- -d - print only duplicated lines

*assumes that the file is sorted

Let’s practice

```
$ sort -n numbers.txt >numbers_sort.txt
$ uniq numbers_sort.txt
$ sort numbers.txt | uniq
$ uniq -d numbers_sort.txt
$ uniq -c numbers_sort.txt
```



```
2 7
```

a line of the file

How many times it occurs

numbers.txt

3
4
5
7
2
1
6
7

Shell scripts

- If you have a bunch of commands you’d like to automate, you can put them on separate lines of a file.

My first shell script!

Following steps are required to write shell script:

- Use any **vi** editor like to write shell script.
- After writing shell script set execute permission for your script

My first shell script!

```
vi script.sh
```

```
#!/bin/bash
echo "My first script”
```

How to run the script :

```
chmod u+x script.sh
./script.sh
```

More scripting

- Let’s create a bash script which will split <gtf> into files corresponding to every chr (2,3,21), save every file in separate directory called chr${i}_gtf.

More scripting

vi script.sh

```
#!/bin/bash
echo "My first script"

mkdir chr2_gtf
mkdir chr3_gtf
mkdir chr21_gtf

grep "chr2\s" $1 >chr2_gtf/chr2.gtf
grep "chr3\s" $1 >chr3_gtf/chr3.gtf
grep "chr21\s" $1 >chr21_gtf/chr21.gtf
```

How to run the script :

```
./script.sh hg19.gtf
```

Environmental modules

- set you environment to be able to run bioinformatics tools

```
module load <modulefile>
```

```
bowtie
BWA
samtools
tophat
```

Let’s practice

```
[serghei@login1 test]$ bwa
-bash: tophat: command not found
[serghei@login1 test]$ module load bwa
[serghei@login1 test]$ bwa
```

How to use qsub

- **qsub** is the command used for job submission to the cluster. It takes several command line arguments and can also use special directives found in the submission scripts or command file.

L

```
qsub -cwd -V -N <name of the proccess> -l
h_data=<MEM_NEEDED>,time=24:00:00 run.sh
```

amount of memory (in megabytes M, or gigabytes G) that your job will require 4GB-32GB

command to run (e.g. map NGS reads)

Let’s practice!

```
. /u/local/Modules/default/init/modules.sh
Module load bwa
bwa mem toy.ref.fastq toy.reads.fastq>toy.reads.bwa.sam
```

← vi script.sh

- `qsub -cwd -V -N testBWA -l h_data=8G,time=1:00:00 run.sh`
- `qstat | grep serghei`

↑

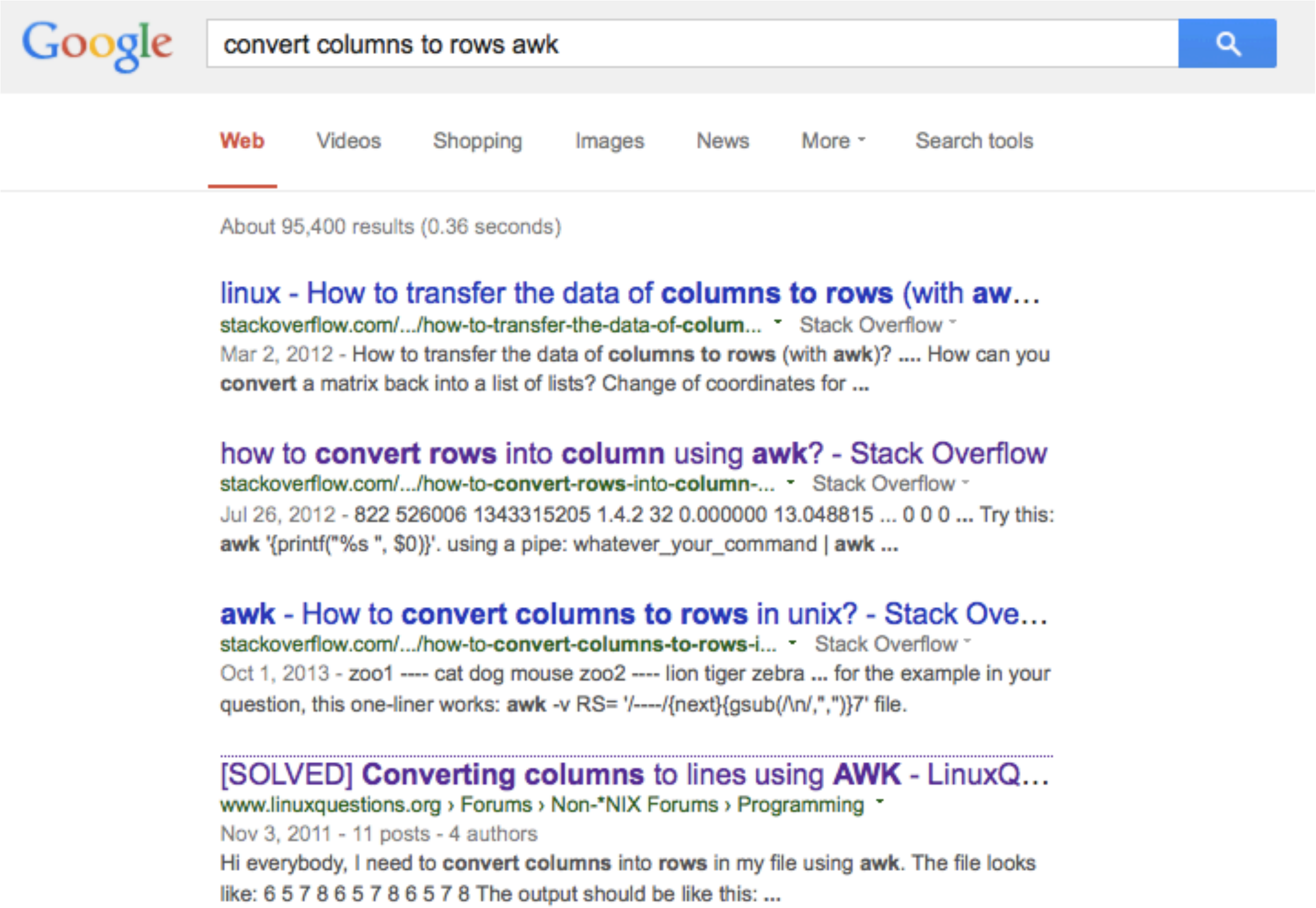
Displays all the jobs which are running on hoffman2

```
toy.reads.bwa.sam:
r1     0     ref    9     60     30M  *     0     0
       ACTGGGGGACTGGGGGTTTTTTTGGACTGG
       ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~   NM:i:0
       MD:Z:30    AS:i:30XS:i:0
```

History

- history | grep awk
- history >history_w1.txt

All UNIX commands from
the workshop in one file



About 95,400 results (0.36 seconds)

linux - How to transfer the data of **columns to rows** (with **aw**…
stackoverflow.com/.../how-to-transfer-the-data-of-colum... Stack Overflow
Mar 2, 2012 - How to transfer the data of **columns to rows** (with **awk**)? .... How can you **convert** a matrix back into a list of lists? Change of coordinates for ...

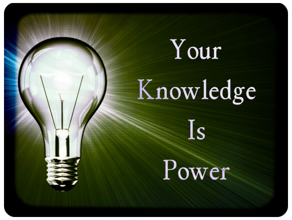
how to **convert rows** into **column** using **awk**? - Stack Overflow
stackoverflow.com/.../how-to-**convert-rows**-into-**column**-... Stack Overflow
Jul 26, 2012 - 822 526006 1343315205 1.4.2 32 0.000000 13.048815 ... 0 0 0 ... Try this: **awk** '{printf("%s ", $0)}'. using a pipe: whatever_your_command | **awk** ...

awk - How to **convert columns to rows** in unix? - Stack Ove…
stackoverflow.com/.../how-to-**convert-columns-to-rows**-i... Stack Overflow
Oct 1, 2013 - zoo1 ---- cat dog mouse zoo2 ---- lion tiger zebra ... for the example in your question, this one-liner works: **awk** -v RS= '/----/{next}{gsub(/\n/,",")}7' file.

[SOLVED] **Converting columns** to lines using **AWK** - LinuxQ…
www.linuxquestions.org › Forums › Non-*NIX Forums › Programming
Nov 3, 2011 - 11 posts - 4 authors
Hi everybody, I need to **convert columns** into **rows** in my file using **awk**. The file looks like: 6 5 7 8 6 5 7 8 6 5 7 8 The output should be like this: ...

http://www.linuxquestions.org/questions/programming-9/converting-columns-to-lines-using-awk-911677/

Do Biologists have to become Programmers?



+

Google

stackoverflow

SEQanswers
the next generation sequencing community

=

*provided in the class

*free and easy to use

ROP

- 🟩 repeat elements
- 🟪 B/T repertoire diversity
- 🟧 circular RNA
- 🟦 microbial communities

https://sergheimangul.wordpress.com/rop/

Thanks!

- Please take a few minutes to fill the survey