Awk Commands ::

-F indicates what type of field separator to use.

i.e. if you want to separate by "," do:

$ awk -F ','

Get rows with KO terms:

awk -F '\t' '$2!=""' AQ6121\_ko.list

Get average size of files in a directory:

ls -l | gawk '{sum += $5; n++;} END {print sum/n;}'

Print the size of the file next to the file name:

awk '{printf ("%s\t%s\n", $5, $9)}' 100k\_Batch2\_nextseq2.txt

Print contig and coverage

4th column is the length

awk '{split($2, a, "\_")}{printf("%s\t%s\n", $1, a[6])}' splitByCov.txt

awk '{split($2, a, "\_")}{printf("%s\t%s\t%s\n", $1, a[4])}' splitByCov.txt

awk '{if ($3 > 10) printf("%s\t%s\t%s\n",$1, $2, $3)}' allContigCov.txt

awk -F '\t' '{print $2}' ATX11195.faa.csv | sort | uniq

Awk to swap two fileds :D

awk ' { t = $1; $1 = $2; $2 = t; print; } ' input\_file

Delete all errored jobs by date

qstat -f | grep "10/01/2017" | awk '{print "qdel " $1}' > del.txt

Delete all jobs in queued waiting:

for i in `qstat -ne -u emfha|grep qw|awk '{print $1}'`; do qdel  $i; done

Gunzip all the reference strains

for i in `ls -d -l /gpfs1/biodb/bacteria\_public\_ncbi\_2018/ncbi-genomes-2018-01-04/\*.gz|awk '{print $9}'`; do gunzip -f $i; done

for i in `ls \*.faa`; do cat $i >> all\_proteomics.faa; done

Delete from the cluster:

qstat -f | grep "35" | awk '{printf("qdel %s\n", $1)}'

Match and split into files by delimiter:

gawk '{if(match($0, /^LOCUS(.+NC\_0[0-9]+)/, k)){name=k[1]}} {print >name".txt" }' Bamy\_references.gb

Basics I

$1

Reference first column

awk '/pattern/ {action}' file↵

Execute action for matched pattern 'pattern' on file 'file'

;

Char to separate two actions

print

Print current record line

$0

Reference current record line

Variables I

$2

Reference second column

FS

Field separator of input file (default whitespace)

NF

Number of fields in current record

NR

Line number of the current record

Basics II

^

Match beginning of field

~

Match opterator

!~

Do not match operator

-F

Command line option to specify input field delimiter

BEGIN

Denotes block executed once at start

END

Denotes block executed once at end

str1 str2

Concat str1 and str2

One-Line Exercises I

awk '{print $1}' file↵

Print first field for each record in file

awk '/regex/' file↵

Print only lines that match regex in file

awk '!/regex/' file↵

Print only lines that do not match regex in file

awk '$2 == "foo"' file↵

Print any line where field 2 is equal to "foo" in file

awk '$2 != "foo"' file↵

Print lines where field 2 is NOT equal to "foo" in file

awk '$1 ~ /regex/' file↵

Print line if field 1 matches regex in file

awk '$1 !~ /regex/' file↵

Print line if field 1 does NOT match regex in file

Variables II

FILENAME

Reference current input file

FNR

Reference number of the current record relative to current input file

OFS

Field separator of the outputted data (default whitespace)

ORS

Record separator of the outputted data (default newline)

RS

Record separator of input file (default newline)

Variables III

CONVFMT

Conversion format used when converting numbers (default %.6g)

SUBSEP

Separates multiple subscripts (default 034)

OFMT

Output format for numbers (default %.6g)

ARGC

Argument count, assignable

ARGV

Argument array, assignable

ENVIRON

Array of environment variables

Functions I

index(s,t)

Position in string s where string t occurs, 0 if not found

length(s)

Length of string s (or $0 if no arg)

rand

Random number between 0 and 1

substr(s,index,len)

Return len-char substring of s that begins at index (counted from 1)

srand

Set seed for rand and return previous seed

int(x)

Truncate x to integer value

Functions II

split(s,a,fs)

Split string s into array a split by fs, returning length of a

match(s,r)

Position in string s where regex r occurs, or 0 if not found

sub(r,t,s)

Substitute t for first occurrence of regex r in string s (or $0 if s not given)

gsub(r,t,s)

Substitute t for all occurrences of regex r in string s

Functions III

system(cmd)

Execute cmd and return exit status

tolower(s)

String s to lowercase

toupper(s)

String s to uppercase

getline

Set $0 to next input record from current input file.

One-Line Exercises II

awk 'NR!=1{print $1}' file↵

Print first field for each record in file excluding the first record

awk 'END{print NR}' file↵

Count lines in file

awk '/foo/{n++}; END {print n+0}' file↵

Print total number of lines that contain foo

awk '{total=total+NF};END{print total}' file↵

Print total number of fields in all lines

awk '/regex/{getline;print}' file↵

Print line immediately after regex, but not line containing regex in file

awk 'length > 32' file↵

Print lines with more than 32 characters in file

awk 'NR==12' file↵

Print line number 12 of file

Sed Commands :

Delete all lines starting with #

sed '/^#/ d'

Convert fastq to fasta:

sed -n '1~4s/^@/>/p;2~4p'

Alternatively, you can convert from fastq to fasta using cutadapt

cutadapt -o output.fasta input.fastq.gz

Pasted from <<http://cutadapt.readthedocs.io/en/stable/guide.html>>

Delete files containing a line:

sed -i '/pattern to match/d' ./infile

Find Commands :

Get the top 10 largest files recursively in a directory

find . -type f -printf "%s\t%p\n" | sort -n | tail -10

Compress all fastqs:

find . -type f -name '\*.fastq' -exec gzip {} \;

Use -maxdepth 1 to look only in the current directory

Find number of files in each subdirectory:

find . -maxdepth 1 -mindepth 1 -type d -exec sh -c 'echo "{} : $(find "{}" -type f | wc -l)" file\(s\)' \;

Use Shell to open multiple files in a list :

Very simple bash command to open all files in a file and check for presence of a string.

for F in $(cat ./Batch14\_cantUpdateBasePathAndHasNoWrittenResources.txt) ; do grep "Failed: Workflow does not have correct modules to run" $F; done;

for F in $(cat ./Batch12\_cantUpdateBasePathAndHasNoWrittenResources.txt) ; do grep "Inserted new rawread entry (rawReadID: '\*') in table rawreads." $F; done;

for F in $(cat allLogs.txt) ; do grep "Traceback" $F; done;

Using the log files, get back all of the commands:

fgrep -f /gpfs2/duds/emfha/Batch14\_cantUpdateBasePathAndHasNoWrittenResources.txt 100k\_Batch14\_150213\_Batch14\_PlateI-00\*\_12\_12.tasks > reRun.tasks

Trying to use bash to get all of the reads md5sum.txt files:

for i in {37..59}; do s3cmd get s3://MOgene\_Dist/ClientUser02/100k\_Batch$i/md5sum.txt ./100k\_Batch$i.md5sum.txt; done;

Pasted from <<https://confluence.be.bayercropscience/display/CKB/How+to+download+reads+from+MoGene>>

Use bash to download all of the reads from the cloud batch by batch

for i in {49..55}; do s3cmd sync s3://MOgene\_Dist/ClientUser02/100k\_Batch$i /gpfs1/MOgene/amazon\_cloud; done;

We need to use the shell to go through the caught errored log files, and see if they all errored only because of the stale file handle situation. Later, we will investigate whether or not the stale file handle is a serious problem.

for F in $(cat ./4\_24\_finalErroredFiles.txt) ; do grep -Hn "Stale file handle" $F; done;

for F in $(cat /gpfs2/duds/emfha/AllLogFilesBatch27Halfway.txt) ; do grep -Hn -B1 "Unable to wait for job because of error: Event client " $F; done;

for F in $(cat /gpfs2/duds/emfha/AllLogFilesBatch27Halfway.txt) ; do grep -Hn -B1 "depthAltCountFile in SNVer was not found" $F; done;

for F in $(cat /gpfs2/duds/emfha/testAllBatch27moduleLogFiles.txt) ; do grep -Hn -B1 "Unable to wait for job because of error: Event client " $F; done;

Use bash to delete all folders

for i in {47..59}; do rm -rf /gpfs1/GAPV3\_10\_2016/100k\_Batch$i; done;

for i in {31..34}; do ls /gpfs1/GAPV3\_10\_2016/100k\_Batch$i/\*.tasks; done;

for i in {2..1661}; do mkdir /gpfs1/workspace/emfha/jira-tickets/GP-1462/output/PUB$i; done;

Grep ::

You can use -A1 to grep also to get the line right after what you matched, and you can use -B1 to get the line right before what you grepped.

Use -H to show the name

Count the number of subdirectories in a directory: ls -l . | grep -c ^d

for i in `awk -F ',' '{print $2}' blasthit\_geneaccession\_short.csv | tr -d '\r'`; do echo $i; cat gene2accession | grep "$i"; done;

Check for contents of one file in another file

grep -F -f file1 file2