

AWK Hints

(The name awk comes from the writers of the program – Aho, Weinberger and Kernighan)

Examples in this document use the file results_fusion_stage2_t2d.csv generated by Anne Jackson in the directory /home/FUSION/Genotypes/QT/Typed/Stage2/ on snowwhite which contains

```
trait,snp,a1,a2,n_eff,dir,beta,se,p_add,chr,pos,reasontrait,gene
for the traits
```

```
bmi, chol, cholhdl, chol_nomed, dbp, dbp_adjbmi, dbp_lt60, dbp_lt60_adjbmi, glu_fast,
glu_fast_adjbmi, hdl, hdl_nomed, hdlratio, height, hip, hip_adjht, homa, homa_adjbmi,
homa_b, homa_b_adjbmi, ins_fast, ins_fast_adjbmi, ldl, ldl_nomed, pp_adjbmi,
pulsepressure, sbp, sbp_adjbmi, tg, tg_nomed, tgratio, trait, waist, waist_adjht, waistht,
waisthtsq, weight, weight_adjht, whr
```

results_fusion_stage2_t2d.csv:

```
trait,snp,a1,a2,n_eff,dir,beta,se,p_add,chr,pos,reasontrait,gene
hdl,rs3764261,2,1,1104,-,-0.252,0.047,9.798E-8,16,55550825,hdl,CETP
hdl_nomed,rs3764261,2,1,924,-,-0.258,0.052,5.804E-7,16,55550825,hdl,CETP
hdl_nomed,rs1864163,3,1,923,+0.254,0.058,0.0000125,16,55554734,hdl,CETP
hdl,rs1864163,3,1,1104,+0.222,0.053,0.00002591,16,55554734,hdl,CETP
hip,rs12986413,1,4,1138,-,-0.159,0.041,0.00009986,19,2121954,height,AP3D1;DOT1L
hdlratio,rs3764261,2,1,1104,-,-0.184,0.047,0.0001051,16,55550825,hdl,CETP
height,rs4743120,2,4,1111,+0.157,0.042,0.0001919,9,97437280,BMI,PTCH1
cholhdl,rs3764261,2,1,1104,+0.176,0.047,0.0001963,16,55550825,hdl,CETP
tg,rs662799,1,3,967,-,-0.325,0.087,0.0002006,11,116168917,tg,APOA5;BUD13;ZNF259
```

trait is field one (\$1), snp is field 2 (\$2), a1 is field 3 (\$3), etc.
\$0 selects an entire line.

Pattern matching similar to grep and Perl can be used.

```
/[Tt]he/ matches The and the
```

```
/[a-z]/ matches any letter from a to z
```

```
/[a-zA-Z0-9]/ matches any letter or number
```

```
/(^bmi) || (^whr)/ matches lines beginning with bmi or with whr (use && for and)
```

```
/^t./ matches lines beginning with tg, tg_nomed, and tgratio
```

```
use \^ or \$ to search for ^ or $
```

awk supports <, <=, ==, !=, >=, and >.

awk uses the standard four arithmetic functions (+, -, *, /). All computations are performed in floating point.

The increment operator is ++; the decrement operator is --.

Using awk from the command line (all output can be directed to a file with ">")

```
awk '/bmi/' infile.csv
```

```
returns all lines containing bmi
```

awk '/^bmi/' infile.csv
returns all lines beginning with bmi

awk '/bmi/{n++;} END { print n+0 }' infile.csv
returns count of lines containing bmi
(same as "grep bmi infile.csv | wc -l")

awk -F, '/bmi/ { print \$1, \$2 }' infile.csv
returns trait and snp where trait contains bmi (bmi_chg_avg, logbmi, etc.)
(default output is space delimited)
-F specifies the input file delimiter (default is space)
-F '\t' specifies tab as the input file delimiter

awk -F, '/bmi/ { OFS = ","; print \$1, \$2 }' infile.csv
returns the same output comma delimited
and
awk -F, '/bmi/ { OFS = "\t"; print \$1, \$2 }' infile.csv
returns the same output tab delimited (could also do { print \$1, "\t", \$2 }
OFS specifies the output file delimiter

awk -F, '\$2 ~ /rs10027062/' infile.csv
returns all lines where the snp (\$2) is rs10027062
(!~ returns all lines where the snp is not rs10027062)
awk -F, '{ if (\$2 ~ /rs10027062/) print }' infile.csv
returns the same information

awk -F, '((NR < 11) && (\$2 ~ /rs3764261/))' infile.csv
returns one line from rows 1-10 containing snp rs3764261

awk -F, '((NR < 11) && (\$2 ~ /rs3764261/) || (\$2 ~ /rs1864163/))' infile.csv
returns the same line for rs3764261 and all lines with rs1864163

awk -F, /"bmi,rs10027062"/ infile.csv
returns all lines containing 'bmi,rs9311811' (string must be in double quotes) which includes
bmi,rs10027062,1,3,1175,+,0.089,0.046,0.05074,4,172681785,whr,na
pp_adjbmi,rs10027062,1,3,1124,+,0.04,0.047,0.3913,4,172681785,whr,na
sbp_adjbmi,rs10027062,1,3,1124,+,0.035,0.047,0.4493,4,172681785,whr,na
homa_adjbmi,rs10027062,1,3,747,+,0.031,0.056,0.5887,4,172681785,whr,na
etc.

awk -F, '{ OFS = ","; print \$9, \$2, \$1 | "sort" }' infile.csv
sorts by p_add and returns p_add, snp, and trait
0.0000125,rs1864163,hdl_nomed
0.00002591,rs1864163,hdl
0.00009986,rs12986413,hip
0.0001051,rs3764261,hdlratio
etc.

```
awk '{ print FNR "\t" $0 }' infile.csv
```

prints the line number for each line of a file (line # is tab delimited)

FNR is the current record number in the current file

Note that the header line is numbered

```
1 trait,snp,a1,a2,n_eff,dir,beta,se,p_add,chr,pos,reasontrait,gene
2 hdl,rs3764261,2,1,1104,-,-0.252,0.047,9.798E-8,16,55550825,hdl,CETP
3 hdl_nomed,rs3764261,2,1,924,-,-0.258,0.052,5.804E-7,16,55550825,hdl,CETP
4 hdl_nomed,rs1864163,3,1,923,+,0.254,0.058,0.0000125,16,55554734,hdl,CETP
```

```
awk '{ printf("%5d : %s\n", NR, $0) }' infile.csv
```

prints the line number for each line specifying 5 characters for the

line number, right aligned, colon delimited

```
1 : trait,snp,a1,a2,n_eff,dir,beta,se,p_add,chr,pos,reasontrait,gene
2 : hdl,rs3764261,2,1,1104,-,-0.252,0.047,9.798E-8,16,55550825,hdl,CETP
3 : hdl_nomed,rs3764261,2,1,924,-,-0.258,0.052,5.804E-7,16,55550825,hdl,CETP
4 : hdl_nomed,rs1864163,3,1,923,+,0.254,0.058,0.0000125,16,55554734,hdl,CETP
```

```
awk -F, '$2 = "Change_Column_2"' infile.csv
```

returns

```
trait Change_Column_2 a1 a2 n_eff dir beta se p_add chr pos reasontrait gene
hdl Change_Column_2 2 1 1104 - -0.252 0.047 9.798E-8 16 55550825 hdl CETP
hdl_nomed Change_Column_2 2 1 924 - -0.258 0.052 5.804E-7 16 55550825 hdl CETP
hdl_nomed Change_Column_2 3 1 923 + 0.254 0.058 0.0000125 16 55554734 hdl CETP
because you never know when you might need this!
```

```
awk '{ print 1, $0 }' infile > outfile
```

adds a column containing 1 to the beginning of each line

```
awk '{ print 20, $0 }' genetic_map_chr20.txt > chr20.txt
```

changes

```
position COMBINED_rate(cM/Mb) Genetic_Map(cM)
```

```
11231 0.734242795126639 0.0010564879985202
```

```
11244 0.734146530138204 0.00106603315485685
```

to

```
20 position COMBINED_rate(cM/Mb) Genetic_Map(cM)
```

```
20 11231 0.734242795126639 0.0010564879985202
```

```
20 11244 0.734146530138204 0.00106603315485685
```

(header line is ignored when uploading to SQL table)

```
awk '{ $1 = ""; print }' infile.csv > outfile.csv
```

removes the first column

```
awk '{ $3 = $4 = ""; print }' infile.csv > outfile.csv
```

removes columns 3 and 4

awk 'NR < 11' infile.csv
returns the first 10 lines of a file (same as head)
NR keeps a current count of the number of input lines

awk '{ print NF }' infile.csv
returns the number of fields in each line of the input file

awk -F, '{ print \$NF }' infile.csv
returns the last field of each line

awk 'END{print}' infile.csv
returns the last line of the file (same as tail -1)

awk 'END {print NR, "rows"}' infile.csv
returns "16227 rows"
(wc -l infile.csv returns
"16227 infile.csv")

wc is easier and makes much more sense :)

awk 'NR==25,NR==30' infile.csv
returns lines 25-30

Using printf

awk -F, '{ if (\$2 ~ /rs10027062/) printf ("%e\n", \$9) }' infile.csv
returns p_add (\$9) in exponential format
1.441000e-02
2.756000e-02

awk -F, '{ if (\$2 ~ /rs10027062/) printf ("%f\n", \$9) }' infile.csv
returns p_add in floating-point format
0.014410
0.027560

awk -F, '{ if (\$2 ~ /rs10027062/) printf ("%g\n", \$9) }' infile.csv
returns p_add in either exponential or floating-point format, whichever is shorter
0.01441
0.02756

awk -F, '{ if (\$2 ~ /rs10027062/) printf ("%2f\n", \$9) }' infile.csv
2 specifies the minimum width or number of spaces the output will use (output will exceed minimum width if it is longer). Output is the same as with %f.

awk -F, '{ if (\$2 ~ /rs10027062/) printf ("[%12f]\n", \$9) }' infile.csv
returns (brackets used for visibility purposes)
[0.014410] right-aligned to 12th space
[0.027560]

```
awk -F, '{ if ( $2 ~ /rs10027062/) printf ("[%-12f]\n", $9) }' infile.csv
returns the same output left justified
[0.014410  ]
[0.027560  ]
```

```
awk -F, '{ if ( $2 ~ /rs10027062/) printf ("%0.3f\n", $9) }' infile.csv
returns the output with 3 decimal places
0.014
0.028
```

```
awk -F, '{ if ( $2 ~ /rs10027062/) printf ("[%8.3f]\n", $9) }' infile.csv
returns the output in 8 spaces with 3 decimal places
[ 0.014]
[ 0.028]
```

Using split

```
awk -F, '{ c=split($0, s); for(n=1; n<=c; ++n) print s[n] }' infile.csv
returns each word/number in a file on a separate line
hdl
rs3764261
2
1
1104
-
-0.252
0.047
9.798E-8
16
55550825
hdl
CETP
```

Using substr (substring)

```
awk -F, '{ print substr($2,1,4) }' infile.csv
returns
the first 4 characters of $2 (snp [not the best example])
```

Create an awk command file to invoke with -f

```
# Example awk program file (traits.awk):
# This is an awk program that lists some of the traits in
# results_fusion_stage2_t2d.csv and their counts
#
/bmi/      { num_bmi++; } # ++ increments value by 1
/chol/     { num_chol++; }
/dbp/      { num_dbp++; }
END {
    printf ("\n");
    print "Counts of Traits in results_fusion_stage2_t2d_alltraits.csv";
    printf ("\n");          # print new line
    printf (" Number of BMI snps:  %d\n", num_bmi);
    printf (" Number of chol snps: %d\n", num_chol);
    printf (" Number of dbp snps:  %d\n", num_dbp);
    printf ("\n");
    printf (" Total number of SNPs:   %d\n", NR);
    printf ("\n");
}
```

```
# Preface comments with #
# printf = print format
# %d = digit/integer number
# END performs the final actions
```

awk -f traits.awk infile.csv returns

Counts of Traits in results_fusion_stage2_t2d_alltraits.csv

```
Number of BMI snps: 3988
Number of chol snps: 1491
Number of dbp snps: 2014
```

```
Total number of SNPs: 16227
```

Other examples

Double-space a single-spaced file

```
awk '1;{ print "" }' infile.csv
```

or

```
awk '{ print ; print "" }' infile.csv
```

Triple-space a single-spaced file

```
awk '1;{ print "\n" }' infile.csv
```

Combine 2 lines into one (input file cannot have a header row):

```
rs1000000,A,1
rs1000000,B,3
rs10000010,A,4
rs10000010,B,2
rs10000023,A,4
rs10000023,B,3
```

```
awk 'ORS=NR%2?" ,":"\n"' infile.csv
```

returns

```
rs1000000,A,1,rs1000000,B,3
rs10000010,A,4,rs10000010,B,2
rs10000023,A,4,rs10000023,B,3
```

ORS joins the lines together with a comma (default is newline)

The Yoruba map file has 2596392 SNPs; 143 are in common with AADM. Use egrep with -n or awk to get SNPs and their line numbers from dat or map file.

lines.txt

```
30340,rs807269
130366,rs6577171
158483,rs2641348
195448,rs3791020
```

The first 5 fields in the pedigree file are famid, studyid, father, mother, and sex. Add 5 to each SNP line number to get the correct genotype column to cut.

```
awk -F, '{ print $1+5 }' lines.txt > numbers.txt
```

returns

```
30345
130371
158488
195453
```

Cut -f1-5,30345,130371,158488,195453, etc. from pedigree file.

JustPushPlay didn't run properly using seq_hdl and seq_tg in place of naffected. 1 needs to be subtracted from agecat_jpp_s1.

```
SEX,STUDYID,famid,agecat_jpp_s1,FATHER,MOTHER,gwa_s1,birthplace,seq_tg
F,0001-402,0001,7,0001?200,0001?300,F1 case,1,
F,0003-100,0003,3,0003?200,0003?300,F1 case,2,
M,0003-500,0003,3,,,SP ctl,8,
M,0004-100,0004,6,0004?200,0004?300,F1 case,13,
F,0004-500,0004,3,,,SP ctl,1,
```

```
awk -F, '{ if ($4 ~/[1-9]/) print $1, $2, $3, $4-1, $5, $6, $7, $8, $9; else print $0; OFS=","; }'
infile.csv
SEX,STUDYID,famid,agecat_jpp_s1,FATHER,MOTHER,gwa_s1,birthplace,seq_tg
F,0001-402,0001,6,0001?200,0001?300,F1 case,1,
F,0003-100,0003,2,0003?200,0003?300,F1 case,2,
M,0003-500,0003,2,,,SP ctl,8,
M,0004-100,0004,5,0004?200,0004?300,F1 case,13,
F,0004-500,0004,2,,,SP ctl,1,
```

Pritzker QC IDs are the same as the original IDs but end with _R1, _R2, and _R3.
 Repair requires 8 distinct character Study IDs.

```
12347,90C04469,F,1,90C04471,90C04470,,F1 case,1
12347,90C04469_R1,F,1,90C04471,90C04470,,,1
12347,90C04469_R2,F,1,90C04471,90C04470,,,2
12347,90C04470,M,1,,,,,1
12347,90C04470_R1,M,1,,,,,1
12347,90C04470_R2,M,1,,,,,2
12347,90C04470_R3,M,1,,,,,2
```

```
awk -F, '{ if ($2 ~ /_R1/) print $1, 1$2, $3, $4, $5, $6, $7, $8, $9; else if ($2 ~ /_R2/)
print $1, 2$2, $3, $4, $5, $6, $7, $8, $9; else if ($2 ~ /_R3/) print $1, 3$2, $3, $4, $5, $6, $7, $8,
$9; else print $0; OFS=","; }' infile.csv
```

```
returns
12347,90C04469,F,1,90C04471,90C04470,,F1 case,1
12347,190C04469_R1,F,1,90C04471,90C04470,,,1
12347,290C04469_R2,F,1,90C04471,90C04470,,,2
12347,90C04470,M,1,,,,,1
12347,190C04470_R1,M,1,,,,,1
12347,290C04470_R2,M,1,,,,,2
12347,390C04470_R3,M,1,,,,,2
```

Useful web pages:

<http://www.calmar.ws/linux/awk.html>
<http://www.ee.ucl.ac.uk/~hamed/misc/awk1line.txt>
<http://www.gnu.org/software/gawk/manual/gawk.html>
<http://www.vectorsite.net/tsawk.html>