sed, AWK, and Bash Scripting

Frederick J Tan
Bioinformatics Research Faculty
Carnegie Institution of Washington, Department of Embryology

2 June 2014

bit.ly/tanlab-teaching

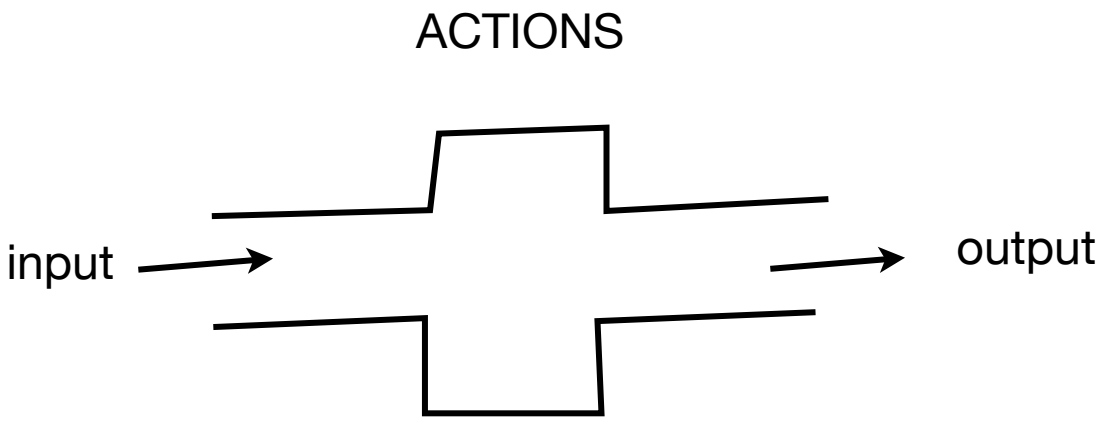
Unix Line-Oriented Programs Help with Many Common Tasks

find and replace (specific patterns of) text

create input file (from differently formatted output file)

summarize (tabular) data

do something again (and again and again and again)




```
cat head tail
grep cut tr
uniq paste join
sed awk
```

A Three Hour Tour

Part I: sed

find and replace
filter by line number

Part II: AWK

convert formats
summarize data

Part III: Bash

automate analysis
increase reproducibility

`sed` Provides Advanced "Find and Replace"

```
$ man sed
```

“stream editor for filtering and transforming text”

COMMON TASKS:

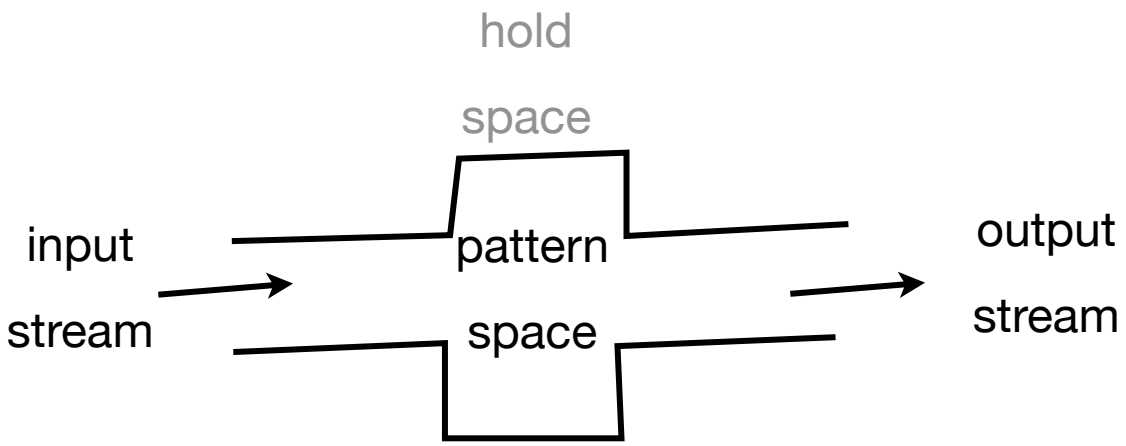
substitute text (i.e. find and replace)

-- WITH regular expressions for pattern matching

-- WITH restrictions to specific lines

... line number, line containing certain text

sed Scripts Follow a General Flow



"**read line** from input and place text in pattern space"

"if find **pattern**, perform **action** on text in pattern space"

(by default) "**print** whatever text is in pattern space"

pattern **action**

`s///` Command Finds and Replaces

```
$ zcat bodymap/data/ERR030876-chr22-001_1.fastq.gz | head
-n12 > reads.fastq
```

```
$ sed '' reads.fastq
```

```
$ sed 's/T/u/' reads.fastq
```

```
$ sed 's/T/u/g' reads.fastq
```

Q: How to *in silico* bisulfite convert CG to tG?

```
$ sed 's/CG/tG/g' reads.fastq
```

Breaking Down sed Patterns and Actions

command	pattern	action
''	anything	nothing
's/T/u/g'	anything	substitute u's for T's

Patterns Can Restrict to a Header or Body

```
$ head bodymap/skeletal.sam > alignments.sam
```

```
$ cat alignments.sam
```

pattern **action**

```
$ sed '1,4s/@/#/' alignments.sam
```

Q: How to change 'C' to 't' from fifth line to the last line?

HINT: google "sed tutorial" and "sed last line"

grymoire.com / "World's best introduction to sed"
stackexchange.com / stackoverflow.com

```
$ sed '5,$s/C/t/g' alignments.sam
```

Breaking Down sed Patterns and Actions

command	pattern	action
''	anything	nothing
's/T/u/g'	anything	substitute u's for T's
'1,4s/@/#/'	lines 1 through 4	substitute # for @

Mid-term sed Quiz

```
$ head bodymap/genes.fpkm_tracking > results.txt
```

Q: How to change 'c' to 'C' on JUST the first line?

```
$ sed '1s/c/C/g' results.txt
```

Q: How to change 'c' to 'C' on every line EXCEPT THE FIRST?

```
$ sed '1!s/c/C/g' results.txt
```

Some Useful sed Patterns and Actions

pattern	action	
1,4	s	substitute
5,$	p	print
1	d	delete
1!	n c g h a ...	
/pattern/		

`sed` Can Simulate `grep`

```
$ sed '/LA16/d' results.txt
```
grep -v

```
$ sed '/LA16/p' results.txt
```

```
$ sed -n '/LA16/p' results.txt
```
grep

Q: What are 3 ways to print only the header lines of alignment.sam?

```
$ sed -n '1,4p' alignments.sam
```

```
$ sed '1,4!d' alignments.sam
```

```
$ sed '5,$d' alignments.sam
```

```
$ sed -n '5,$!p' alignments.sam
```

```
$ sed -n '/@/p' alignments.sam
```

```
$ sed '/@/!d' alignments.sam
```

Breaking Down sed Patterns and Actions

command	pattern	action
''	anything	nothing
's/T/u/g'	anything	substitute u's for T's
'1,4s/@/#/'	lines 1 through 4	substitute # for @
-n '/LA16/p'	text "LA16"	print

`~step` Filters for Every nth Line

Q: How to convert .fastq to .fasta format?

`first~step`

```
$ sed -n '1~4p; 2~4p' reads.fastq
```

```
>@ERR030876.10/1
CTAATTTTTGTAATTTTAGTAGAGACAGGGGTTCTCCATGGGGGCAAGGC
```

Some Useful Regular Expressions

\t	tab	`$ echo -e "apple\torange" | sed -n '/\t/p'`
\n	newline	`$ echo -e "apple orange" | sed -n '/\t/p'`

\s \d \w

[ATCG]	A T C or G	.	any character
[^ATCG]	any character not ATCG	+ *	1 or more / 0 or more
[a-z0-9]	any lowercase or number	{3}	exactly 3 times
[[:alpha:]]	any letter	{3,6}	3 to 6 times
[[:digit::]]	any number	^ $	start / end of line
[[:alnum:]]	any letter or number	|	OR
[[:space:]]	space, tab, newline	( ) and \1	matched pattern

NOTE on metacharacters (e.g. **| \ { }**):

single quotes (') require escaping with '****' to **activate**

double quotes (") require escaping with '****' to **suppress**

e.g. `$ echo "apple orange" | sed -n '/p\{1,2\}/p'`

`sed` Can Quickly Convert .fastq to .fasta

Q: How to convert .fastq to .fasta format?

`first~step`

```
$ sed -n '1~4p; 2~4p' reads.fastq
```

```
>@ERR030876.10/1
CTAATTTTTGTAATTTTAGTAGAGACAGGGGTTCTCCATGGGGGCAAGGC
```

```
$ sed -n '1~4s/^/>/; 1~4p; 2~4p' reads.fastq
```

```
$ sed -n '1~4{s/^/>/;p}; 2~4p' reads.fastq
```

Breaking Down sed Patterns and Actions

command	pattern	action
''	anything	nothing
's/T/u/g'	anything	substitute u's for T's
'1,4s/@/#/'	lines 1 through 4	substitute # for @
-n '/LA16/p'	text "LA16"	print
-n '1~4p; 2~4p'	lines 1+(4n) and lines 2+(4n)	print

Final sed Quiz

Q: How to show second read in reads.fastq?

```
$ sed -n '5,8p' reads.fastq
```

Q: How to convert all quality scores <30 to 0?

```
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJ
0.2.....................................41
                                    ^--Q30
```

```
$ sed '4~4s/[^?@ABCDEFGHIJ]/#/g' reads.fastq
```

Some of the Many Things `sed` Can Do

Convert DNA to RNA; *in silico* bisulfite convert CpGs

Convert .fastq to .fasta format

Manipulate table/.SAM header/non-header lines

Print only specific lines

Print only lines that do/don't contain a pattern

A Three Hour Tour

Part I: sed

find and replace
filter by line number

Part II: AWK

convert formats
summarize data

Part III: Bash

automate analysis
increase reproducibility

`awk` Provides Format Conversions and Summaries

```
$ man awk
```

“pattern scanning and text processing language"
... created by Aho, Weinberger, and Kernighan

COMMON TASKS:

substitute text (i.e. find and replace)

-- WITH regular expressions for pattern matching

-- WITH restrictions to specific lines

convert data formats

-- WITH automatic column splitting

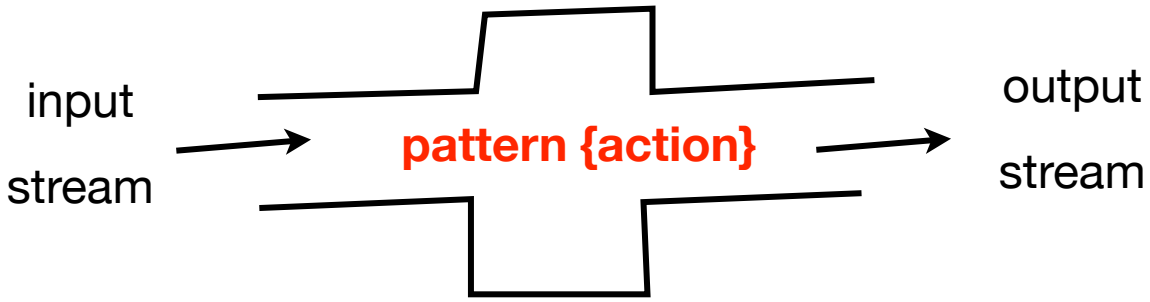
-- WITH string manipulations

summarize data

-- WITH built-in math functions

-- WITH associative arrays

AWK Scripts Flow Slightly Differently Than sed Scripts



“**read line** from input, **split** into columns"	“if find **pattern**, perform **action(s)**"	(by default) "do nothing else"

```
$ awk '' reads.fastq
```

AWK Scripts Can Be Longer Than Comparable sed Scripts

sed command	AWK command
''	'{print}'
's/T/u/g'	'{gsub(/T/,"u"); print}'
'1s/@/#/'	'NR==1 {gsub(/@/,"#")} {print}'
-n '/LA16/p'	'/LA16/'
-n '1~4p; 2~4p'	'NR%4==1 || NR%4==2'

Input Lines Are Automatically Parsed Input into Records and Fields

records ($0)

tracking_id	gene_id	gene_short_name	locus
ENSG00000229286	ENSG00000229286	LA16c-4G1.4	chr22:16076051-16076172
ENSG00000233866	ENSG00000233866	LA16c-4G1.3	chr22:16062156-16063236
ENSG00000235265	ENSG00000235265	LA16c-4G1.5	chr22:16084248-16084826
ENSG00000223875	ENSG00000223875	NBEAP3	chr22:16100516-16124973
ENSG00000215270	ENSG00000215270	LA16c-60H5.7	chr22:16122719-16123768
ENSG00000230643	ENSG00000230643	LA16c-60G3.5	chr22:16389484-16389602
ENSG00000234381	ENSG00000234381	LA16c-59E1.2	chr22:16333632-16342783
ENSG00000224435	ENSG00000224435	NF1P6	chr22:16345911-16355362
ENSG00000231565	ENSG00000231565	NEK2P2	chr22:16364866-16366204

$1 $2 $3 $4

fields

Anything `cut` Can Do `awk` Can Do (Better)

```
$ ls -l
```

```
-rw-rw-r-- 1 workshop workshop 1.8K May 30 15:25 alignments.sam
drwxrwxr-x 2 workshop workshop 4.0K May 30 14:16 bin
drwxrwxr-x 3 workshop workshop 4.0K May 30 14:19 bodymap
drwxrwxr-x 3 workshop workshop 4.0K May 30 14:16 genomes
drwxrwxr-x 2 workshop workshop 4.0K May 30 14:17 PacBio
-rw-rw-r-- 1 workshop workshop  360 May 30 15:24 reads.fastq
-rw-rw-r-- 1 workshop workshop  954 May 30 15:26 results.txt
drwxrwxr-x 2 workshop workshop 4.0K May 30 14:17 RM11
drwxrwxr-x 5 workshop workshop 4.0K May 30 14:14 src
```

```
$ ls -l | awk '{print $9, $5}'
```

`awk` Can Substitute More Safely Than `sed`

```
$ head genomes/human/hg19-chr22-UCSC-knownGene.bed >
annotations.bed
```

```
$ awk '{gsub(/^chr/,"",$1); print}' annotations.bed
```

sub() gsub()

print() printf()

length() split()

substr() match() index()

tolower() toupper()

Create a .bed File, Converting from 1-based to 0-based Starts

```
$ head genomes/human/hg19-chr22-iGenomes.gtf >
annotations.gtf
```

```
$ awk '{print $1, $4-1, $5, $10}' annotations.gtf
```

+ - * /

log() sqrt() exp()

Differences with Unix cut:

-- reorder columns

-- modify fields

-- default delimiter [\t]+

... potential to skip empty cells

Change Field Delimiters By Modifying FS and OFS

	FS (input) field separator	OFS output field separator
default	[\t]+	[ ]
cut style	[\t]+	[\t]+
.csv format	[,]	[,]

```
$ awk '/^[^@]/ {print $1,$3,$4}' alignments.sam
```

```
$ awk '/^[^@]/ {print $1,$3,$4}' OFS="\t" alignments.sam
```

```
$ awk '/^[^@]/ {OFS="\t"; print $1,$3,$4}' alignments.sam
```

The BEGIN Block Is Executed Before Any Input Is Processed

BEGIN {action} **pattern {action}** END {action}

```
$ awk 'BEGIN {OFS="\t"} /^[^@]/ {print $1,$3,$4}'
alignments.sam
```

Q: How to add header line(s)?

```
$ awk 'BEGIN {print "one two three"} {print}' file.txt
```

The END Block Is Useful for Outputting Summaries

```
$ awk '{total=total+$3-$2} END {print total}'
annotations.bed
```

Mid-term AWK Quiz

Q: How to extend .bed annotations +/- 50 bp?

```
$ awk '{print $1, $2-50, $3+50}' annotations.bed
```

Q: How to create separate entries for -50bp and +50bp ?

```
$ awk '{print $1, $2-50, $2; print $1, $3, $3+50}'
annotations.bed
```

Create a Histogram Using Associative Arrays

Q: How many mutations found per chromosome?

```
$ grep -v ^# RM11/var.raw.vcf | awk '
/chrI/ {chrI++} /chrII/ {chrII++} /chrIII/ {chrIII++} ...
/chrV/ {chrV++} /chrVI/ {chrVI++} /chrVII/ {chrVII++} ...
END{print "chrI",chrI; print "chrII",chrII; ...}'
```

count["chrI"]++	count["chrV"]++
count["chrII"]++	count["chrVI"]++
count["chrIII"]++	count["chrVII"]++
...	...

```
$ grep -v ^# RM11/var.raw.vcf | awk '
{count[$1]++}
END{for (i in count) print i,count[i]}'
```

Track Number of Records and Fields with the NR and NF Variables

```
$ awk 'NR%4==1 || NR%4==2' reads.fastq
```

&& ||

> < == != >= <=

~ !~

```
$ less -S bodymap/skeletal.sam
```

```
$ awk '{fields[NF]++} END{for (i in fields) print
i,fields[i]}' bodymap/skeletal.sam
```

Final AWK Quiz

Q: How to create .fastq file with PacBio reads >6,000 bases?

```
$ awk '{a=$0;getline;b=$0;getline;c=$0;getline;print
a"\t"b"\t"c"\t"$0}' PacBio/m<TAB> > pacbio.tab
```

```
$ awk 'length($2) > 6000 {print $1; print $2; print $3;
print $4}' pacbio.tab > pacbio.6kb.fastq
```

Some of the Many Things `awk` Can Do

Convert chromosome names from UCSC to Ensembl (chrI to I)

Convert .gtf to .bed format

Add header line(s)

Summarize total size of annotations

Modify .bed annotations

Create histogram of features per chromosome

Convert .fastq to table format

A Three Hour Tour

Part I: sed

find and replace
filter by line number

Part II: AWK

convert formats
summarize data

Part III: Bash

automate analysis
increase reproducibility

Setting and Unsetting Shell Variables

```
$ name="Frederick Tan"
```

```
$ echo name
```

```
$ echo $name
```

```
$ set | grep ^name
```

```
$ unset name
```

```
$ echo $name
```

Your First Bash Script

```
$ nano doAnalysis.sh
```

```
name="Frederick Tan"
echo $name
```

<CTRL>-X
to save and exit

```
$ chmod +x doAnalysis.sh
```

```
$ ./doAnalysis.sh
```

```
$ echo $name
```

```
$ which bash
```

```
$ nano doAnalysis.sh
```

```
#!/bin/bash
name="Frederick Tan"
echo $name
```

Passing Command Line Arguments

```
$ nano doAnalysis.sh
```

```
#!/bin/bash
name=$1
echo $name
```

```
$ ./doAnalysis.sh "Frederick Tan"
```

```
$ ./doAnalysis.sh
```

```
$ nano doAnalysis.sh
```

```
#!/bin/bash
name=$1
if [ "$1" == "" ]; then
  echo "Please supply a name"
  exit
fi
echo $name
```

<-- spaces are important!

A Real Pre-Processing Script

```
$ nano doAnalysis.sh
```

```
#!/bin/bash

zcat /home/workshop/bodymap/data/ERR030876-
chr22-001_1.fastq.gz > ERR030876-chr22-001_1.fastq

sed -n "1~4p;2~4p" ERR030876-chr22-001_1.fastq > ERR030876-
chr22-001_1.fasta
```

```
$ ./doAnalysis.sh
```

Throw in Five More Datasets

```
$ nano doAnalysis.sh
```

```
#!/bin/bash
zcat /home/workshop/bodymap/data/ERR030876-chr22-001_1.fastq.gz > ERR030876-chr22-001_1.fastq
sed -n "1~4p;2~4p" ERR030876-chr22-001_1.fastq > ERR030876-chr22-001_1.fasta
zcat /home/workshop/bodymap/data/ERR030876-chr22-001_2.fastq.gz > ERR030876-chr22-001_2.fastq
sed -n "1~4p;2~4p" ERR030876-chr22-001_2.fastq > ERR030876-chr22-001_2.fasta
zcat /home/workshop/bodymap/data/ERR030876-chr22-002_1.fastq.gz > ERR030876-chr22-002_1.fastq
sed -n "1~4p;2~4p" ERR030876-chr22-002_1.fastq > ERR030876-chr22-002_1.fasta
zcat /home/workshop/bodymap/data/ERR030876-chr22-002_2.fastq.gz > ERR030876-chr22-002_2.fastq
sed -n "1~4p;2~4p" ERR030876-chr22-002_2.fastq > ERR030876-chr22-002_2.fasta
zcat /home/workshop/bodymap/data/ERR030876-chr22-003_1.fastq.gz > ERR030876-chr22-003_1.fastq
sed -n "1~4p;2~4p" ERR030876-chr22-003_1.fastq > ERR030876-chr22-003_1.fasta
zcat /home/workshop/bodymap/data/ERR030876-chr22-003_2.fastq.gz > ERR030876-chr22-003_2.fastq
sed -n "1~4p;2~4p" ERR030876-chr22-003_2.fastq > ERR030876-chr22-003_2.fasta
```

Q: How would you generalize?

Generalize Sample Names

```
$ nano doAnalysis.sh
```

```
#!/bin/bash
for sample in ERR030876-chr22-001_1 ERR030876-chr22-001_2
ERR030876-chr22-002_1 ERR030876-chr22-002_2 ERR030876-
chr22-003_1 ERR030876-chr22-003_2
do
  zcat /home/workshop/bodymap/data/$sample.fastq.gz >
$sample.fastq
  sed -n "1~4p;2~4p" $sample.fastq > $sample.fasta
done
```

Generalize More

```
$ nano doAnalysis.sh
```

```
#!/bin/bash
DATA_DIR="/home/workshop/bodymap/data/"
SAMPLES="ERR030876-chr22-001_1 ERR030876-chr22-001_2
ERR030876-chr22-002_1 ERR030876-chr22-002_2 ERR030876-
chr22-003_1 ERR030876-chr22-003_2"

for sample in $SAMPLES
do
  zcat $DATA_DIR/$sample.fastq.gz > $sample.fastq
  sed -n "1~4p;2~4p" $sample.fastq > $sample.fasta
done
```

Generalize Even More

```
$ nano doAnalysis.sh
```

```
#!/bin/bash
SED="/bin/sed"
SED_SCRIPT="-n '1~4p;2~4p'"

DATA_DIR="/home/workshop/bodymap/data/"
SAMPLES="ERR030876-chr22-001_1 ERR030876-chr22-001_2
ERR030876-chr22-002_1 ERR030876-chr22-002_2 ERR030876-
chr22-003_1 ERR030876-chr22-003_2"

for sample in $SAMPLES
do
  zcat $DATA_DIR/$sample.fastq.gz > $sample.fastq
  eval $SED $SED_SCRIPT $sample.fastq > $sample.fasta
done
```

Check If File Already Exists

```
$ rm *.fasta
```

```
$ nano doAnalysis.sh
```

```
#!/bin/bash
SED="/bin/sed"
SED_SCRIPT="-n '1~4p;2~4p'"

DATA_DIR="/home/workshop/bodymap/data/"
SAMPLES="ERR030876-chr22-001_1 ERR030876-chr22-001_2
ERR030876-chr22-002_1 ERR030876-chr22-002_2 ERR030876-
chr22-003_1 ERR030876-chr22-003_2"

for sample in $SAMPLES
do
  if [ ! -f $sample.fastq ]; then
    echo "Making $sample.fastq"
    zcat $DATA_DIR/$sample.fastq.gz > $sample.fastq
  fi
  if [ ! -f $sample.fasta ]; then
    echo "Making $sample.fasta"
    eval $SED $SED_SCRIPT $sample.fastq > $sample.fasta
  fi
done
```

Build File Lists with Brace Expansion

```
$ echo color{Red,Orange,Yellow}
```

```
$ cp bodymap/data/ERR030876-chr22-00{1_1,1_2}.fastq.gz .
```

```
$ SAMPLE_PREFIX="ERR030876-chr22-00"
```

```
$ SAMPLE_SUFFIX="1_1,1_2,2_1,2_2,3_1,3_2"
```

```
$ SAMPLES=$SAMPLE_PREFIX{$SAMPLE_SUFFIX}
```

```
$ eval cp bodymap/data/$SAMPLES.fastq.gz .
```

Some of the Many Ways to Bash Script

Document workflow

Declare settings in one place

Minimize redundancy with variables

Accept command-line arguments

Check if files exist

Enumerate with brace expansion

Shell parameter expansion

Command substitution

Floating point math

Arrays

Useful Websites

sed	grymoire.com/Unix/sed.html catonmat.net/blog/worlds-best-introduction-to-sed
AWK	grymoire.com/Unix/Awk.html
Bash	gnu.org/software/bash/manual/html_node/index.html
Regular expressions	grymoire.com/Unix/Regular.html
Command line questions and tricks	stackoverflow.com superuser.com commandlinefu.com
Workshop notes	bit.ly/tanlab-teaching