



# H3ABioNet

Pan African Bioinformatics Network for H3Africa

## Introduction to SED

Sumir Panji / Amel Ghouila

Material adapted from: H3ABioNet and WT NGS Bioinformatics Course Africa 2021

# Stream Editor (SED)

- You can find patterns in files using grep
- What happens if you want to change some characters or patterns?
- What happens if you only want to print certain lines in a file to use for another software program or to check?
- Can be done with SED allows one to this quite easily
- Find SED very useful for finding, substituting and formatting text and files e.g. fasta headers

# Basic SED syntax

- sed 's/pattern to find/pattern to replace/' input\_file
- The s after sed in the command is for substitution
- E.g. sed 's/chr/Chromosome/' practical/Notebooks/awk/genes.gff
- sed has a couple of default ways of working:
  - sed reads in the file looks for matches of the pattern line by line
  - The output is sent to the standard out / screen line by line
- The output of the above command provides:

Chromosome1	source1	gene	100	300	0.5	+	0
name=gene1;product=unknown							
Chromosome1	source2	gene	1000	1100	0.9	-	0
name=recA;product=RecA protein							
Chromosome1	source5	repeat	10000	14000	1	+	.
name=ALU							
Chromosome2	source2	gene	10000	1200	0.95	+	0

# Basic SED syntax

- Fantastic!!! But a stream of characters my screen does not help me as I need an actual modified file for use
- One generally redirects the output from the screen to a new file
- `sed 's/pattern to find/pattern to replace/' input_file > output_file`
- E.g. `sed 's/chr/Chromosome/' practical/Notebooks/awk/genes.gff > sed_output_genes.gff`

# Basic SED syntax

- The previous example is a bit misleading for a reason:
- Say I need to format the genes.gff file from its current tab-delimited format to a comma separated one for another program
- `sed 's/\t/,/' practical/Notebooks/awk/genes.gff`

```
chr1,source1      gene      100      300      0.5      +      0
                  name=gene1;product=unknown
chr1,source2      gene      1000     1100     0.9      -      0
                  name=recA;product=RecA protein
chr1,source5      repeat    10000    14000    1        +      .
                  name=ALU
```

# Basic SED syntax

- Only the first tab was substituted by a comma, and not the rest of the tabs
- Why?

# Basic SED syntax

- Only the first tab was substituted by a comma, and not the rest
- Why?
- Recall SED works by reading lines and matching to the first pattern it finds in the line
- For the sed 's/chr/Chromosome/' example – chr appears once on each new line
- For the sed 's/\t/,/' – the tab character appears multiple times on each new line
- sed's default behaviour is to substitute the first match on each new line

# Basic SED syntax

- What if we want to replace all the matches to the pattern regardless of the number of times it appears in a new line?
- Use the global flag
- E.g. `sed 's/\t/,/g' practical/Notebooks/awk/genes.gff`

`chr1,source1,gene,100,300,0.5,+,0,name=gene1;product=unknown`

`chr1,source2,gene,1000,1100,0.9,-,0,name=recA;product=RecA protein`

`chr1,source5,repeat,10000,14000,1,+,,name=ALU`

`chr2,source2,gene,10000,1200,0.95,+,0`



# Counting and extracting lines with SED

- One can use sed to print out specific lines in a file e.g a row
- Say I wanted to extract lines 1, 2 and 3 only from the genes.gff file
- `sed '1,3p' practical/Notebooks/awk/genes.gff`
- Notice that the substitution flag and slashes are not present as we are just extracting lines, not matching and modifying any characters

# Counting and extracting lines with SED

sed '1,3p' practical/Notebooks/awk/genes.gff

chr1	source1	gene	100	300	0.5	+	0	
	name=gene1;product=unknown							
chr1	source1	gene	100	300	0.5	+	0	
	name=gene1;product=unknown							
chr1	source2	gene	1000	1100	0.9	-	0	
	name=recA;product=RecA protein							
chr1	source2	gene	1000	1100	0.9	-	0	
	name=recA;product=RecA protein							
chr1	source5	repeat	10000	14000	1	+	.	name=ALU
chr1	source5	repeat	10000	14000	1	+	.	name=ALU
chr2	source2	gene	10000	1200	0.95	+	0	
chr2	source1	gene	50	900	0.4	-	0	
	name=gene2;product=gene2 protein							
chr3	source1	gene	200	210	0.8	.	0	name=gene3
chr4	source3	repeat	300	400	1	+	.	name=ALU
chr10	source2	repeat	60	70	0.78	+	.	name=LINE1
chr10	source2	repeat	150	166	0.84	+	.	name=LINE2
chrX	source1	gene	123	456	0.6	+	0	
	name=gene4;product=unknown							

# Counting and extracting lines with SED

- In this case sed printed out the whole file and added lines 1 and then 3 within the file
- Why?

# Counting and extracting lines with SED

- Recall sed's default behavior is to print everything out onto the screen
- We can use the -n option to prevent sed's default behaviour of printing everything to the screen
- E.g sed -n '1,3p' practical/Notebooks/awk/genes.gff

chr1	source1	gene	100	300	0.5	+	0	
		name=gene1;product=unknown						
chr1	source2	gene	1000	1100	0.9	-	0	
		name=recA;product=RecA protein						
chr1	source5	repeat	10000	14000	1	+	.	name=ALU

# Counting and extracting lines with SED

- “sed -n ‘1,3,p’ practical/Notebooks/awk/genes.gff” prints out a range of lines from 1 to 3
- How do I get it to print out specific lines e.g 1-3 and then 5 and 7
- sed -n ‘1,3p; 5p; 7p’ practical/Notebooks/awk/genes.gff

chr1	source1	gene	100	300	0.5	+	0	
								name=gene1;product=unknown
chr1	source2	gene	1000	1100	0.9	-	0	
								name=recA;product=RecA protein
chr1	source5	repeat	10000	14000	1	+	.	name=ALU
chr2	source1	gene	50	900	0.4	-	0	
								name=gene2;product=gene2 protein
chr4	source3	repeat	300	400	1	+	.	name=ALU

# Special characters for SED

- I mainly use sed for pattern matching and substitution and formatting of files
- Sed provides a number of useful characters to provide more control over its pattern matching:
  - ^ match the start of the line
  - \$ match the end of the line
  - [a-z] characters of the alphabet – used to change cases using the U& (for upper case) and L& for lower case (note can also use the unix command tr for case changing)
  - sed -n 'p;n' – print out odd number lines
  - sed -n 'n;p' – print out even number lines

# Special characters for SED

- `sed 's/^/Organism_/g' genes.gff`

Organism_chr1	source1	gene	100	300	0.5	+	0
name=gene1;product=unknown							
Organism_chr1	source2	gene	1000	1100	0.9	-	0
name=recA;product=RecA protein							

- `sed 's/$/_Organism/g' genes.gff`

chr1	source1	gene	100	300	0.5	+	0
name=gene1;product=unknown_Organism							
chr1	source2	gene	1000	1100	0.9	-	0
name=recA;product=RecA protein_Organism							

- `sed 's/[a-z]/\U&/g' genes.gff`

- |                            |         |      |     |     |     |   |   |
|----------------------------|---------|------|-----|-----|-----|---|---|
| CHR1                       | SOURCE1 | GENE | 100 | 300 | 0.5 | + | 0 |
| NAME=GENE1;PRODUCT=UNKNOWN |         |      |     |     |     |   |   |
- |                                |         |      |      |      |     |   |   |
|--------------------------------|---------|------|------|------|-----|---|---|
| CHR1                           | SOURCE2 | GENE | 1000 | 1100 | 0.9 | - | 0 |
| NAME=RECA;PRODUCT=RECA PROTEIN |         |      |      |      |     |   |   |

# Special characters for SED

- For more specific control, can use the pattern to be matched e.g.
- `sed 's/^chr*/Organism_/g' genes.gff`

Organism_chr1	source1	gene	100	300	0.5	+	0
name=gene1;product=unknown							
Organism_chr1	source2	gene	1000	1100	0.9	-	0
name=recA;product=RecA protein							

- `sed 's/$/_Organism/g' genes.gff`

chr1	source1	gene	100	300	0.5	+	0
name=gene1;product=unknown_Organism							
chr1	source2	gene	1000	1100	0.9	-	0
name=recA;product=RecA protein_Organism							

- `sed 's/[a-z]/\U&/g' genes.gff`

- CHR1 SOURCE1 GENE 100 300 0.5 + 0  
NAME=GENE1;PRODUCT=UNKNOWN
- CHR1 SOURCE2 GENE 1000 1100 0.9 - 0  
NAME=RECA;PRODUCT=RECA PROTEIN



# More info and examples on using SED (syntaxes / usage might differ)

- <https://bioinformaticsworkbook.org/Appendix/Unix/unix-basics-4sed.html#gsc.tab=0>
- <https://dasher.wustl.edu/chem478/software/unix-tools/sed.html>
- <https://www.grymoire.com/Unix/Sed.html>
- <https://gist.github.com/ssstonebraker/6140154>

# Thank you!!!

**Acknowledgements: Amel Ghouila and course material adapted from  
H3ABioNet WT NGS Bioinformatics Course Africa 2021**