Searching & Redirection

Dr. Princess Rodriguez

2025-01-22

Learning objectives

- Search for characters or patterns in a text file using the `grep` command
- Write to and append a file using output redirection
- Use of pipe (`|`) character
  - How can I combine existing commands to do new things?

Searching files with `grep` command

In the same way that many of us now use 'Google' as a verb meaning 'to find', UNIX programmers often use the word `grep`.

- `grep` is a contraction of '**g**lobal/**r**egular **e**xpression/**p**rint', a common sequence of operations in early UNIX text editors. It is also the name of a very useful command-line program.
- `grep` allows you to search plain-text files without opening them.

The syntax for `grep` is as follows:

```
grep  search-term  filename
```

Other useful options for grep

This will limit matches to word boundaries.

```
grep -w
```

Sometimes we don't want to search for a single word, but for a phrase. We can also do this by putting the phrase in quotes:

```
grep "is not" haiku.txt
```

Another useful option is `-n` which numbers the lines that match:

```
grep -n "it" haiku.txt
```

More information about the FASTQ file format

Line	*Description*
1	*Read name preceded by '@'*
2	*The actual DNA sequence*
3	*Read name (same as line 1) preceded by a '+' or just a '+' sign*
4	*String of characters which represent the quality score of each nucleotide in line 2; must have same number of characters as line 2*

Redirection

Redirection allows us to send the output from the Terminal to another destination. In this case, we can save the output to a file, which lets us examine it at our convenience.

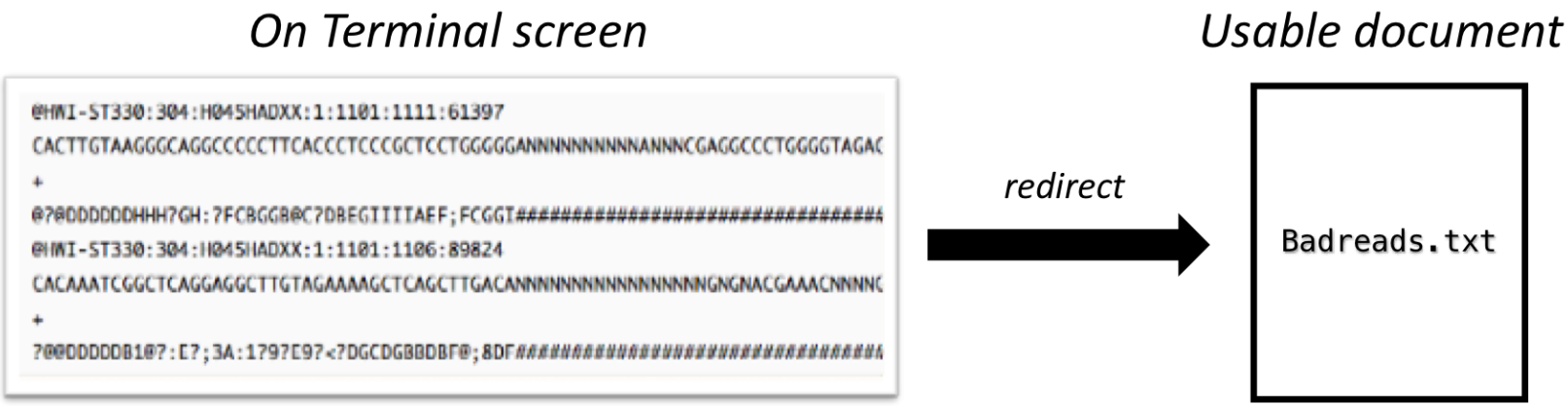

Figure 1: Redirection

Redirecting with > AKA "Greater-than sign"

The redirection command for writing something into a file is >.

Let's put all the sequences that contain `NNNNNNNNNN` from the `Mov10_oe_1.subset.fq` into another file called `bad_reads.txt`.

```
grep -B 1 -A 2 NNNNNNNNNN Mov10_oe_1.subset.fq > bad_read
```

Redirecting (and appending) with >>

The redirection command for appending something to an existing file is >>.

If we use >> it will **append** to the existing content in a file rather than overwrite it. This can be useful for saving more than one search. For example, the following command will append the bad reads from **Mov10_oe_2** to the `bad_reads.txt` file that we just generated.

```
grep -B 1 -A 2 NNNNNNNNNN Mov10_oe_2.subset.fq >> bad_re
ls -l
```

Explaining Appending

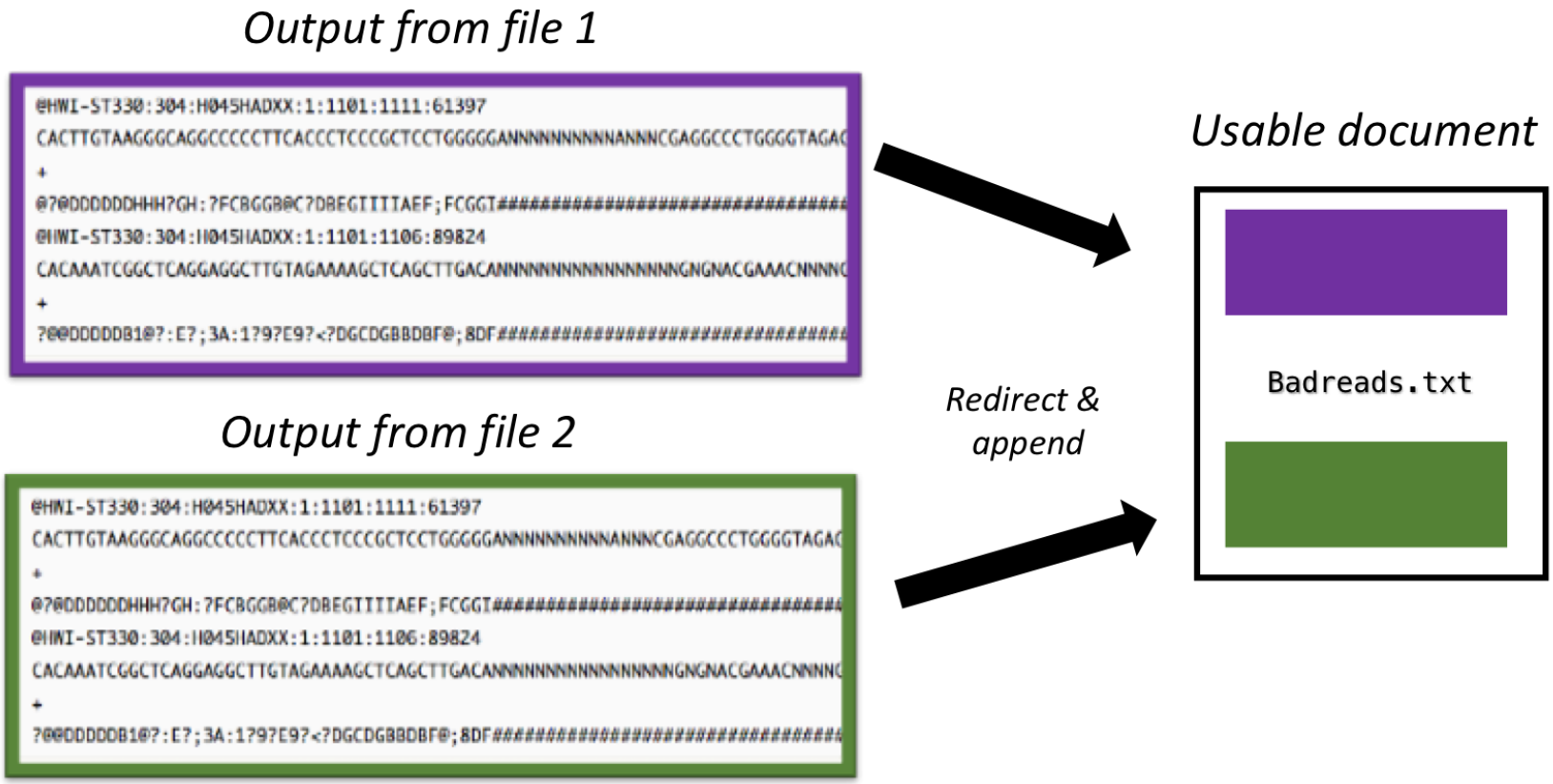

Figure 2: Appending

Passing output to another command with | (or pipe)

What | does is take the output from one command and runs it through the command specified after it.

```
grep -B 1 -A 2 NNNNNNNNNN Mov10_oe_1.subset.fq | head -n
```

The Power of Pipes

- You can string along as many commands together as you like

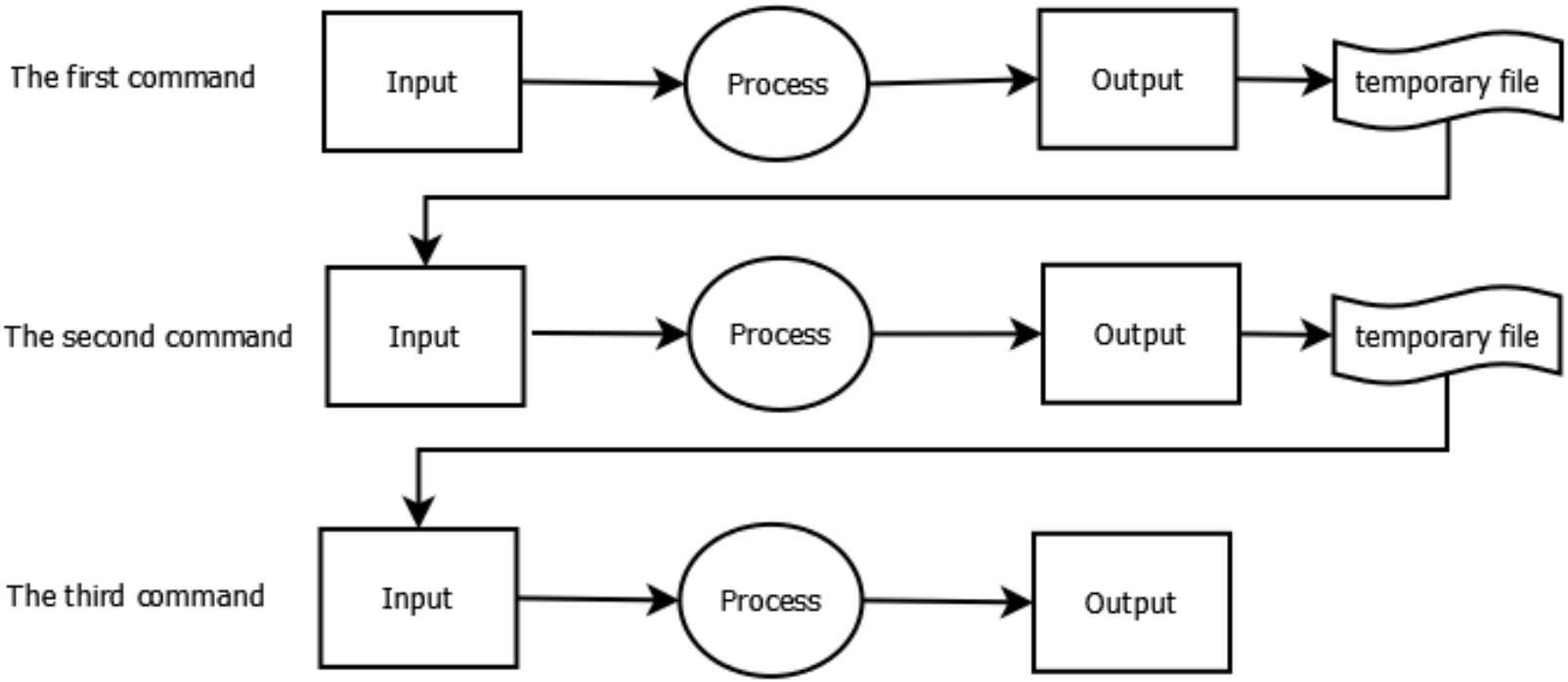

Figure 3: Power of Pipes

Introducing the GTF file format

Line	Description
1	chromosome number
2	source, name of program that generated the feature - its “unknown” above
3	feature type name
4	start position of feature
5	end position of feature
6	score
7	strand, defined at + (forward) or - (reverse)
8	frame
9	attribute, provides additional information about each feature

Cut & Sort

- `cut` **is a command that extracts columns from files.**
- `sort` **is a command used to sort the contents of a file in a particular order.** It has arguments that let you pick which column to sort by (`-k`), what kind of sorting you want to do (numeric `n`) and also if the result of the sorting should only return unique (`-u`) values. These are just 2 of the many features of the sort command.

Summary

```
grep            # Allows for searching within files without
                 +   grep search_term filename

>               # Redirect output to another file

>>              # append to an existing file rather than ov

|               # Pipe key
                  + takes the output and runs it through th

cut             # used to extract specific columns from a t

sort            # used to sort a specific column within a t
```

Citation

This lesson has been developed by members of the teaching team at the Harvard Chan Bioinformatics Core (HBC). These are open access materials distributed under the terms of the Creative Commons Attribution license (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

- *The materials used in this lesson were derived from work that is Copyright © Data Carpentry (http://datacarpentry.org/). All Data Carpentry instructional material is made available under the Creative Commons Attribution license (CC BY 4.0).*
- *Adapted from the lesson by Tracy Teal. Contributors: Paul Wilson, Milad Fatenejad, Sasha Wood, and Radhika Khetani for Software Carpentry (http://software-carpentry.org/)*
- *Original Authors: Sheldon McKay, Bob Freeman, Mary Piper, Radhika Khetani, Meeta Mistry, Jihe Liu, Will Gammerdinger*