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?

Recap from last week

- ∗ matches zero or more characters in a filename
- ? matches a single character in a filename
- Use of the control key may be described in many ways, including Ctrl-X, Control-X, or ˆX.
- The shell does not have a trash bin, once something is deleted it is really gone.
- Most file names are **something.extension**. The extension is not required, but does help bioinformatic programs find required files for downstream processing.
- Depending on the type of work you do, you may need a more powerful text editor than Nano.

Class Exercise: Participation Grade

https://docs.google.com/forms/d/e/1FAIpQLSdgFp-9H4lH3QaK9Z8MLHkhl1ydXlzQxtI6XZcydAjw8VE1XA/viewform

Searching files with grep

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
```

Class Exercise #2: haiku.txt

Class Exercise Discussion

In the above exercise, *not* was the first pattern we were searching for. The `grep` command searches through the file, and then looks for matches to the pattern specified.

- `grep` searches for a pattern in a *case-sensitive way*.
- Finally, you would have saw upon completing #6, the search pattern we have selected *does not have to form a complete word or phrase* as the following where two results:
  - The Tao
  - My **The**sis

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:

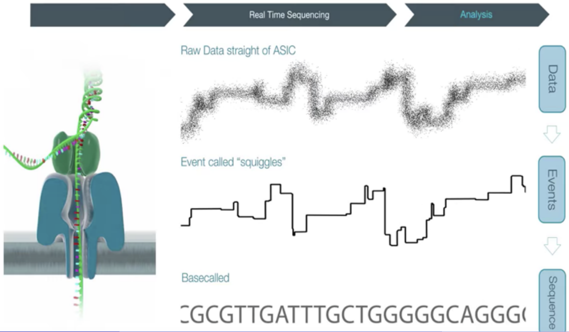
```
grep "is not" haiku.txt
```

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

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

Next-Generation Sequencing (NGS): Base calling

The Next-Generation Sequencing (NGS) technologies all rely on a complex interplay of chemistry, hardware and optical sensors. Adding to this complexity is software to analyze the sensor data to predict the individual bases. This is called **basecalling**.

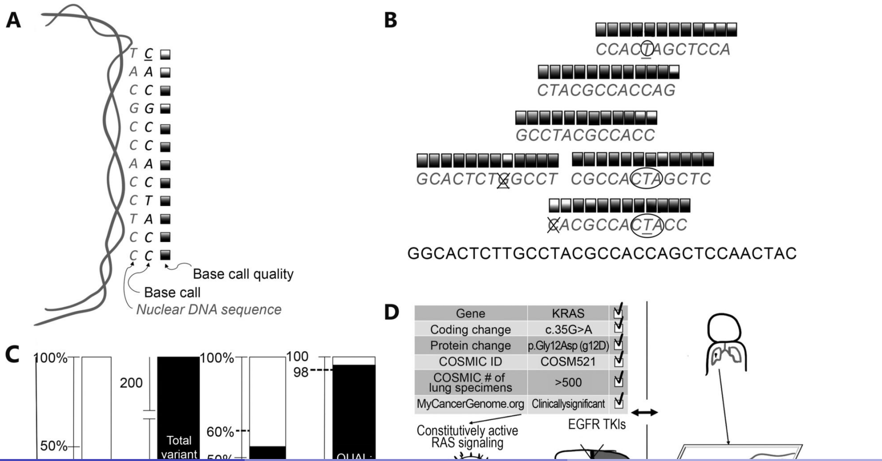


What is Basecalling?

- Basecalling is the process of translating electrical signals from a sequencing device into a DNA or RNA sequence.
- Base-calling algorithms process the raw signal to decode the sequence of bases within strands of DNA or RNA into data stored in BAM or FASTQ files.
- A Phred score is a quality score that measures the probabiliy of an incorrect base call.

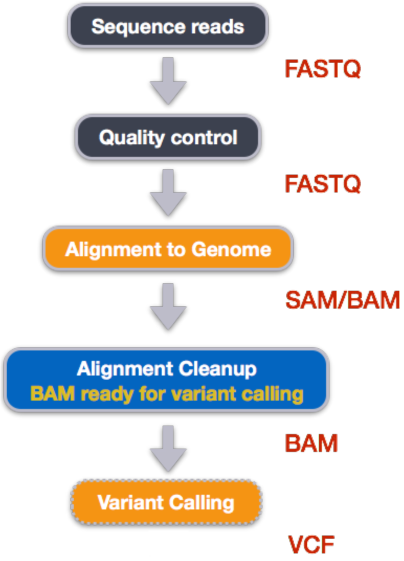
Example: Importance of Basecalling in Variant Analysis

Below is figure from Cancer Biology & Medicine



Gene	KRAS ✓
Coding change	c.35G>A ✓
Protein change	p.Gly12Asp (g12D) ✓
COSMIC ID	COSM521 ✓
COSMIC # of lung specimens	>500 ✓
MyCancerGenome.org	Clinicallysignificant ✓

Variant Calling Workflow



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*

Question for Class:

So what happens if the sequencer is unable to make a decision on which base (A, G, C, T) is the correct one?

Answer

The sequencer will designate an **N**.

Class Exercise #3

Suppose we want to see how many reads in `Mov10_oe_1.subset.fq` contain "bad" data. Use `grep` to find how many reads contain 10 consecutive Ns (`NNNNNNNNNN`) in this file.

Go ahead and perform the command for this.

Class Exercise #4

- Use `grep` to search for the sequence CTCAATGAGCCA in `Mov10_oe_1.subset.fq`. How many sequences did you find and which line number did they appear in?
- How can you modify the command you typed above so that your search also returns the name/header of the sequence?
- If you want to search for the above sequence in **all** Mov10 replicate fastq files, what command would you use?

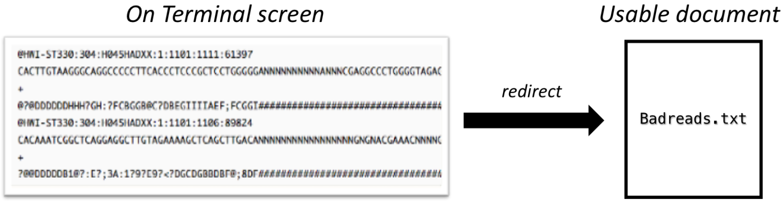
Redirection Prompt

When we use grep, the matching lines are displayed directly in the Terminal (also known as Standard Output or stdout). If the search results contain only a few lines, we can easily read them. However, if the output is very long, the lines will scroll rapidly, and we'll only be able to see the last few lines on the screen.

To avoid losing valuable information and to review the results later, we can save the output to a file instead of letting it scroll past in the Terminal.

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 the contents at our convenience.



```
@HWI-ST330:304:H045HADXX:1:1101:1111:61397
CACTTGTAAGGGCAGGCCCCCTTCACCCTCCCGCTCCTGGGGGANNNNNNNNNNANNNCGAGGCCCTGGGGTAGAC
+
@?@DDDDDDHHH?GH:?FCBGGB@C?DBEGIIIIAEF;FCGGT#################################
@HWI-ST330:304:H045HADXX:1:1101:1106:89824
CACAAATCGGCTCAGGAGGCTTGTAGAAAAGCTCAGCTTGACANNNNNNNNNNNNNNNNNNGNGNACGAAACNNNNC
+
?@@DDDDDB1@?:E?;3A:1?9?E9?<?DGCDGBBDBF@;8DF#################################
```

Figure 4: 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_reads.txt
```

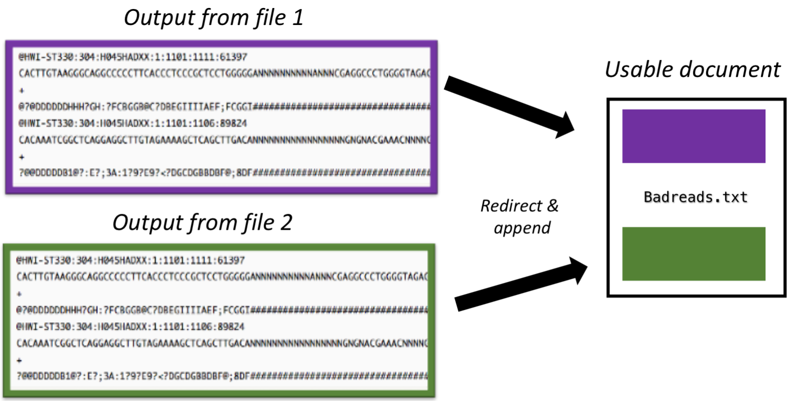
Redirecting (and appending) with >>

- The redirection command for appending something to an existing file is >>.
- For example, the following command will append the "bad reads" from **Mov10_oe_2**/the second biological replicate into the `bad_reads.txt` file that we just generated.

```
grep -B 1 -A 2 NNNNNNNNNN Mov10_oe_2.subset.fq
>> bad_reads.txt

ls -l
```

Explaining Appending



```
@HWI-ST330:304:H045HADXX:1:1101:1111:61397
CACTTGTAAGGGCAGGCCCCCTTCACCCTCCCGCTCCTGGGGGANNNNNNNNNNANNNCGAGGCCCTGGGGTAGAC
+
@?@DDDDDDHHH?GH:?FCBGGB@C?DBEGIIIIAEF;FCGGT####################################
@HWI-ST330:304:H045HADXX:1:1101:1106:89824
CACAAATCGGCTCAGGAGGCTTGTAGAAAAGCTCAGCTTGACANNNNNNNNNNNNNNNNNGNGNACGAAACNNNNC
+
?@@DDDDDB1@?:E?;3A:1?9?E9?<?DGCDGBBDBF@;8DF####################################
```

```
@HWI-ST330:304:H045HADXX:1:1101:1111:61397
CACTTGTAAGGGCAGGCCCCCTTCACCCTCCCGCTCCTGGGGGANNNNNNNNNNANNNCGAGGCCCTGGGGTAGAC
+
@?@DDDDDDHHH?GH:?FCBGGB@C?DBEGIIIIAEF;FCGGT####################################
@HWI-ST330:304:H045HADXX:1:1101:1106:89824
CACAAATCGGCTCAGGAGGCTTGTAGAAAAGCTCAGCTTGACANNNNNNNNNNNNNNNNNGNGNACGAAACNNNNC
+
?@@DDDDDB1@?:E?;3A:1?9?E9?<?DGCDGBBDBF@;8DF####################################
```

Figure 5: 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 5
```

The Power of Pipes

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

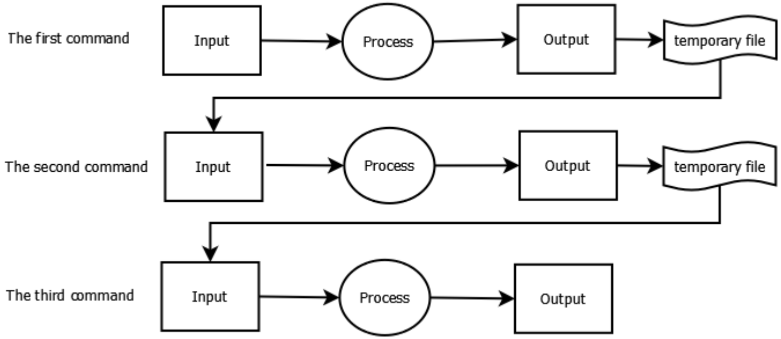


Figure 6: Power of Pipes

Summary for Redirection operators

- The philosophy behind the three redirection operators ($>$, », $|$) you have learned so far is that none of them by themselves do a lot. BUT when you start chaining them together, you can do some really powerful things really efficiently.
- To be able to use the shell effectively, becoming proficient in the use of the pipe and redirection operators is essential.

Introducing the GTF file formal

A GTF file, which stands for “Gene Transfer Format,” is a text-based file format used in bioinformatics to store detailed information about gene structures, including their location on a chromosome, different features like exons and introns, and other relevant attributes, essentially providing a comprehensive annotation of a genome at the gene level

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

Class Exercise #5: Investigating transcripts

- Dr. Patel: Emma, I'd like you to investigate whether the gene PLEKHN1 has two or more transcripts annotated in our GTF file. Understanding this will help us better interpret its potential isoforms in our RNA-seq data.
- Emma: Sure! Could you remind me how to approach this?

Dr. Patel: Absolutely. Here's what you need to do:

- 1. Start with the GTF file. The file chr1-hg19_genes.gtf is located in the reference data folder.
- 2. Search for PLEKHN1. Use the grep command to pull out all the lines related to PLEKHN1, then pipe the output to less.
- 3. Focus on transcript IDs (9th column). Is the same transcript ID being shown?

Class Exercise #6: Identifying genomic coordinates

Dr. Patel: Emma, now that you've identified the transcripts for PLEKHN1, let's dig a bit deeper. I'd like you to extract the genomic coordinates for this gene. Specifically, I need the chromosome, start, and end positions. Use the following:

```
cut -f 1,2,3
```

cut is a command that extracts columns from files, -f = field, and 1,2,3 correspond to the columns of interest.

Please save the final file as *coordinates_PLEKHN1.txt*

`cut`, `sort` & `uniq`

- **`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.
- **`uniq` is a command used to filter out repeated lines in a file.** It's commonly used in conjunction with `sort` to remove duplicate lines from a file, as `uniq` only detects adjacent duplicates

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*