Week 2, Lecture 4

István Albert
Bioinformatics Consulting Center
Penn State
Shell scripts

Collect multiple commands into a single program

• This allows you to the same commands at another time or other data

• Document the steps and describe the thought process
The shell is a programming environment

Execute it with: `sh commands.sh`

(there are other ways to execute shell scripts)
Make your script more informative

```bash
# this is just to generate an empty line
echo

# number of lines in the file
echo The number of lines in the file is:
wq -l data.fastq

# this is just to generate an empty line
echo

# find a pattern in the sequences
echo The number of times the motif ATGC was found is:
grep ATGC < data.fastq | wc -l
```
Running the shell script

```
$ sh commands.sh
The number of lines in the file is:
1000000 data.fastq
The number of times the motif ATGC appears is:
74756
$
```

This is not entirely correct, the quality string may also have the motif in it although with far less chance.

From now on print out your scripts for the homework whenever possible.

Add the description into the script as comments – that is the best place to keep it.
Bash has lots of features

We will slowly introduce some features along the way
Fastx toolkit

Introduction
The FASTX-Toolkit is a collection of command line tools for Short-Reads FASTA/FASTQ files preprocessing.

Next-Generation sequencing machines usually produce FASTA or FASTQ files, containing multiple short-reads sequences (possibly with quality information).

The main processing of such FASTA/FASTQ files is mapping (aka aligning) the sequences to reference genomes or other databases using specialized programs. Example of such mapping programs are Bio, SHRIMP, LastZ, MAQ and many many others.

However,
It is sometimes more productive to preprocess the FASTA/FASTQ files before mapping the sequences to the genome - manipulating the sequences to produce better mapping results.

The FASTX-Toolkit tools perform some of these preprocessing tasks.

Available Tools

Install it by the instructions sent out via the email.
Tip → auto-completion: type a small part of the name fast then quickly press TAB twice
Get help with the –h flag


Part of FASTX Toolkit 0.0.13 by A. Gordon (gordon@cshl.edu)

[-h] = This helpful help screen.
[-f N] = First base to keep. Default is 1 (=first base).
[-l N] = Last base to keep. Default is entire read.
[-t N] = Trim N nucleotides from the end of the read.
        '-t' can not be used with '-l' and '-f'.
[-m MINLEN] = With [-t], discard reads shorter than MINLEN.
[-z] = Compress output with GZIP.
[-i INFILE] = FASTA/Q input file. default is STDIN.
[-o OUTFILE] = FASTA/Q output file. default is STDOUT.
Generate a quality plot

```bash
$ # generate the statistics data
$ fastx_quality_stats.exe -Q 33 -i data.fastq -o stats.txt
$ # generate a boxplot
$ fastq_quality_boxplot_graph.sh -i stats.txt -o quality.png
```

Generated with fastx

Generated with fastqc
Quality trim

```
$ fastq_quality_filter.exe -Q 33 -q 40 -p 50 -i data.fastq > better.fastq

$ wc -l better.fastq
314388 better.fastq

$ wc -l data.fastq
1000000 data.fastq
```

before filtering

![Quality score distribution over all sequences before filtering](image)

after filtering

![Quality score distribution over all sequences after filtering](image)
Data/Tool forensics

• What went wrong with the experiment, sequencing or analysis?

• What is wrong with this tool?

For example on cygwin I get this:
It appears that on Cygwin the last line is not flushed properly

• we can circumvent this problem by a workaround that temporally fixes the problem

  (broadly this type of fixes are called **monkey-patching** everyone will at some point have to stare down a problem like this)

• Only we had a way to keep all but the last line of a file ...

  look at the manual for the command **head**
We’ll do it again

• In solidarity with to our Cygwin users Mac users please cut out the last 4 lines.

```bash
$ fastq_quality_filter.exe -Q 33 -q 40 -p 50 -i data.fastq | head -n -1 > better.fastq
$ wc -l better.fastq
314388 better.fastq
$
Use the **data.fastq** file from lecture 3. Print out the shell scripts that do the following:

- Create a shell script that filters the data.fastq file so that the resulting file:
  1. contains only sequences that do not have polyA/T sites (chains of As or Ts that are at least 6 bp long) *(clipping)*
  2. rename each fasta long header to a number *(renaming)*
  3. filter it so at least 50% of the reads have a quality of 40 *(quality filter)*
  4. generate a FASTA file out of the results *(fastq_to_fasta)*

Due to the cygwin bug you may need to cut off the end of a few lines. Experiment and check the end of the files.