Introduction To NGS Data & Analytic Tools

Steve Pederson

Bioinformatics Hub
University of Adelaide

March 30, 2016

Licensing

Contents

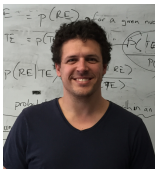
Workshop Information

The Trainers

Mr. Steve Pederson
Co-ordinator
Bioinformatics Hub
The University of Adelaide
South Australia
stephen.pederson@adelaide.edu.au

Dr. Terry Bertozzi
Bioinformatician
SA Museum
South Australia
Terry.Bertozzi@samuseum.sa.gov.au

Dr. Hien To
Bioinformatician
Bioinformatics Hub
The University of Adelaide
South Australia
hien.to@adelaide.edu.au



Dr. Jimmy Breen
Bioinformatician
Bioinformatics Hub & Robinson Research Institute
The University of Adelaide
South Australia
jimmy.breen@adelaide.edu.au

Welcome

Thank you for your attendance & welcome to the Introduction to NGS Data & NGS Analytic Tools Workshop. This is a free offering by the University of Adelaide, Bioinformatics Hub which is a centrally funded initiative from the Department of Vice-Chancellor (Research), with the aim of assisting & enabling researchers in their work. Training workshops & seminars such as this one are an important part of this initiative.

The Bioinformatics Hub itself has a web-page on the University website at http://www.adelaide.edu.au/bioinformatics-hub/. To be kept up to date on upcoming events and workshops, please join the internal Bioinformatics mailing list on http://list.adelaide.edu.au/mailman/listinfo/bioinfo. Also, please consider following the Bioinformatics Hub on Twitter as many handy hints are posted on this feed https://twitter.com/UofABioinfoHub.

Today's workshop has been prepared with generous technical support & advice provided by Dr Nathan Watson-Haigh (*ACPFG*), Dr Dan Kortschak (*Adelaide University, Adelson Research Group*), Dr Zhipeng Qu (*Adelaide University, Adelson Research Group*), Dr Mark Corbett (*Robinson Institute*) & Dr John Toubia (*Adelaide Centre for Cancer Genomics*). The tutors today are Steve Pederson, Dr Hien To, Dr Jimmy Breen from the University of Adelaide, Bioinformatics Hub, & Dr Terry Bertozzi (*SA Museum & Adelaide University, Adelson Research Group*). We hope it will be useful in enabling you to continue and to advance your research.

Course Summary

In today's workshop we will be introducing you to a small number of the basic tools required for NGS data handling, as well as giving you a basic familiarity with what the data actually looks like. Whilst we will not be able to cover all of the rich & diverse set of tools available, we hope to cover many of the key concepts & questions to ask of your data, as well as give you an understanding of what information is actually in the data.

The majority of data handling and analysis required in the field of bioinformatics uses the *command line*, alternatively known as the terminal or the *bash shell*, so some useful tips for the command line will be included amongst the material. Whilst most of the session will involve looking at individual files, in reality most of our analysis will be performed using some type of script to automate, & easily reproduce an analysis.

Today's session is also intended to explore several tools in actual detail, rather than rush across the whole field. There is large amount of information that we won't have time to discuss, but hopefully some important tools and thought processes will be covered &

enable you make better progress with your own datasets.

Post Workshop

The VMs which we work on today will remain active for week or so after the workshop, so feel free to continue exploring any sections that you weren't able to make it through. We'd also encourage you to sign up for some of the high-traffic websites like *BioStars* or *SEQanswers* as these are a rich resource for your own problem solving.

Document Structure

We have provided you with a printed and an electronic copy of the workshop's hands-on tutorial documents. We have done this for two reasons: 1) you will have something to take away with you at the end of the workshop, and 2) you can save time (mis)typing commands on the command line by using copy-and-paste.

We advise you to use Acrobat Reader to view the PDF. This is because it properly supports some features we have implemented to ensure that copy-and-paste of commands works as expected. This includes the appropriate copy-and-paste of special characters like tilde and hyphens as well as skipping line numbers for easy copy-and-past of whole code blocks.

While you could fly through the hands-on sessions doing copy-and-paste, you will learn more if you use the time saved from not having to type all those commands, to understand what each command is doing!

The commands to enter at a terminal look something like this:

```
1 tophat --solexa-quals -g 2 --library-type fr-unstranded -j \
    annotation/Danio_rerio.Zv9.66.spliceSites -o tophat/ZV9_2cells \
    genome/ZV9 data/2cells_1.fastq data/2cells_2.fastq
```

The following styled code is not to be entered at a terminal, it is simply to show you the syntax of the command. You must use your own judgement to substitute in the correct arguments, options, filenames etc

```
tophat [options]* <index_base> <reads_1> <reads_2>
```

The following icons are used in the margin, throughout the documentation to help you navigate around the document more easily:


Important


For reference


Follow these steps


Questions to answer


Warning - STOP and read


Bonus exercise for fast learners


Advanced exercise for super-fast learners

Computer Setup

We will all be working on our own computers today, and will be accessing Virtual Machines running the Ubuntu operating system on the Nectar Research Cloud (http://nectar.org.au/). The software client No Machine which you will have already installed, enables us to access these machines in a familiar Desktop style, even though the majority of our time will be spent within the terminal.

If you did not install NoMachine prior to today's workshop, the correct installers can be found at:

- http://tinyurl.com/lcn3ns8 (Windows)
- http://tinyurl.com/mnm23ew (Mac)

No Machine session files will be provided to each attendee which have been pre-configured to enable easy access to these machines. These machines are effectively dual-core machine with 8GB of RAM & 70GB of hard drive space. Whilst relatively small, this will be more than enough to become familiar with the important concepts for the day. To begin today's session, simply click (or double-click) on the No Machine session file that you have been given. The desktop from the Virtual Machine (VM) that you have connected to will appear on the No Machine client.

While this is connecting, select the button circled in blue above. This will resize NoMachine to be full screen to feel more like you are physically located at the VM. If any security warnings appear, ignore them & continue connecting. This may take several attempts and if you can't connect after 3 or 4 times, call a tutor over.

If you forgot to set the client to full screen, hover your mouse over the top-right corner and click where the 'page' folds over. This will enable you to resize the client again, and can also be used to disconnect from the VM.

The Ubuntu Desktop

Now that you are connected, you will notice we are in a standard graphical environment. The default Desktop in Ubuntu is Unity, but what we are seeing is known as the Gnome Desktop, as is the default for Nectar VMs. As many of us are used to seeing, there are click-able icons on the desktop, and drop-down menus.

The main interfaces we will be using today are the `terminal`, a text editor named `gedit` and the web browser `firefox` They will appear as icons on the desktop, but can also be accessed from the drop-down menus. A `Firefox` icon is also located on the desktop & this can be used for searching for answers as well as viewing the output of `fastqc`.

Basic Shell Comands

Today we will assume a basic familiarity with the bash terminal in Linux. Some key commands which we will be using are explained in the following table.

Command	Meaning
`cd`	Change directory
`ls`	List files in a directory
`man`	Call the manual for a given command
`head`	View the first 10 lines of a file

Open a terminal in the VM by either clicking on the icon, or using the drop-down menu. It will automatically open in your home directory (`/home/trainee`), which is indicated by the tilde before the dollar sign. The tilde (~) is a symbolic representation of the address `/home/trainee`, where the first slash is the root of the computerâĂŹs file system. To return to this directory from any other in the file system, you can simply enter the command

```
1 cd
```

Normally we specify the directory to change into after the command `cd`, but when no directory is given, this command will automatically change to your home directory.

Please note, the Linux operating system is case-sensitive. When entering the any commands, please make sure that you pay attention to this detail. Spaces are an important feature as well, so be careful not to add or omit them where they appear in any sample code.

NGS Data & FASTQ Format

Primary Author(s):
Steve Pederson, Bioinformatics Hub, University of Adelaide
stephen.pederson@adelaide.edu.au

Contributor(s):
Dan Kortschak, Adelson Research Group, University of Adelaide
dan.kortschak@adelaide.edu.au
Terry Bertozzi, SA Museum Terry.Bertozzi@samuseum.sa.gov.au

Initial Goals

1. Understand the *Sequencing by Synthesis* Process & Data Generation

2. Understand what errors and artefacts lie within the data

3. Learn how to assess the quality of data & make informed decisions

NGS Data Generation

Before we can begin to analyse any data, it is helpful to understand how it was generated. Whilst there are numerous platforms for generation of NGS data, today we will look at the Illumina *Sequencing by Synthesis* method, which is one of the most common methods in use today. Many of you will be familiar with the process involved, but it may be worth looking at the following 5-minute video from Illumina: http://youtu.be/womKfikWlxM

This video refers to the process *tagmentation*. This is a relatively recent method for fragmenting & attaching adaptors to DNA, with an alternative, more traditional method being sonication, poly-adenylation & attachment of appropriate adaptors in separate steps. The important concept to note during sample preparation is that the DNA insert has multiple sequences ligated to either end. These include 1) the *sequencing primers*, 2) index or *barcode* sequences, and 3) the flow-cell binding oligos.

Assuming each 'spot' on the flowcell is generated from a unique DNA sequence, there are two important sequencing errors that will occur during this process. What do you think they might be?

Will these errors have a more significant effect if they occur during the sequence detection stage, or during generation of the DNA clones within each cluster?

Will all of the individual ssDNA molecules within a spot be amplified perfectly in sync with each other?

Today's Data

The raw data for today's workshop needs to be downloaded and placed in the directory `/home/trainee/NGSData`. First we'll have to change into our home directory and create this new sub-directory

```
cd
mkdir NGSData
```

There are four main datasets we'll look at today and we'll download them all as we need them. The first dataset today is some RNA-Seq reads, so let's create a directory to place the data in

```
cd ~/NGSData
mkdir -p RNASeq/rawData
```

What did the `-p` option do in the previous line?
This was a quick & dirty shortcut. Why might this be a bad idea?

Now we have somewhere to put our data, let's download it. This is probably the best time to use cut & paste from the electronic copy of these notes. When in the `bash shell` we can copy using `Ctrl + Shift + V` and we can paste using `Ctrl + Shift + V`. Also note that for the second line, you can use the up arrow and just change the end of the line from `reads1.fq.gz` to `reads2.fq.gz`.

```
1 cd RNASeq/rawData
2 wget \
     https://swift.rc.nectar.org.au:8888/v1/AUTH_33065ff5c34a4652aa2fefb292b3195a/IntroNGS
3 wget \
     https://swift.rc.nectar.org.au:8888/v1/AUTH_33065ff5c34a4652aa2fefb292b3195a/IntroNGS
```

FASTQ File Format

As the sequences are extended during the sequencing reaction, an image is recorded which is effectively a movie or series of frames at which the addition of bases is recorded & detected. We mostly don't deal with these image files, but will handle data generated from these in *fastq* format, which can commonly have the file suffix *.fq* or *.fastq*. As these files are often very large, they will often be zipped using `gzip` or `bzip`. Whilst we would instinctively want to unzip these files using the command `gunzip`, most NGS tools are able to work with zipped fastq files, and this is usually unnecessary. This can save considerable hard drive space, which is an important consideration when handling NGS datasets as the quantity of data can easily push your storage capacity to it's limit.

We should still have a terminal open from the previous section &, if necessary, use the `cd` command to make sure you are in the `~/NGSData/RNASeq/rawData` directory. The command `zcat` unzips a file & prints the output to the terminal, or standard output (*stdout*). If we did this to these files, we would see a stream of data whizzing past in the terminal, but instead we can just pipe the output of `zcat` to the command head to view the first 10 lines of a file.

```
cd ~/NGSData/RNASeq/rawData
zcat reads1.fq.gz | head -n8
```

In the above command, we have used a trick commonly used in Linux systems where we have taken the output of one command (`zcat reads1.fq.gz`) and sent it to another command (`head`) by using the *pipe symbol* (—). This is literally like sticking a pipe on the end of a process & redirecting the output to the input another process. If you think of things as being like a data factory you can almost visualise it. There are no limits to the number of commands that you can string together using this trick. Additionally, we gave the argument `-n8` to the command `head` to ensure that we only printed the first eight lines.

> **Don't Panic!!!**
> If at some stage today you find that the terminal has become unresponsive, or you are seeing an unexpected stream of data fly past, you can abort whichever process is currently running in the terminal by entering `Ctrl-c`. This is an instruction to the computer to 'kill the current process.' & you may be surprised at how often this comes in handy. Even experienced programmers rely on this trick from time to time.

In the output from the above terminal command, we have obtained the first 8 lines of the gzipped fastq file. This gives a clear view of the fastq file format, where each individual read spans four lines. These lines are:

1. The read identifier
2. The sequence read
3. An alternate line for the identifier (commonly left blank as just a `+` symbol acting as a placeholder)
4. The quality scores for each position along the read as a series of ascii text characters.

Let's have a brief look at each of these lines and what they mean.

The read identifier

This line begins with an `@` symbol and although there is some variability, it traditionally has several components. Today's data have been sourced from an EBI data repository with the identifier `SRR031714`. For the first sequence in this file, we have the full identifier `@SRR031714.1 HWI-EAS299_130MNEAAXX:2:1:785:591/1` which has the following components:

`SRR031714.1`	The aforementioned EBI identifier & the sequence ID within the file. As this is the first read, we have the number 1. NB: This identifier is **not** present when data is obtained directly from the machine or service provider.
`WHI-EAS299_130MNEAAXX`	The unique machine ID
`2`	The flowcell lane
`1`	The tile within the flowcell lane
`785`	The x-coordinate of the cluster within the tile
`591`	The y-coordinate of the cluster within the tile
`/1`	Indicates that this is the first read in a set of paired end reads

As seen in the subsequent sections, these pieces of information can be helpful in identifying if any spatial effects have affected the quality of the reads. By and large you won't need to utilise most of this information, but it can be handy for times of serious data exploration.

While we are inspecting our data, have a look at the beginning of the second file.

```
zcat reads2.fq.gz | head -n8
```

Here you will notice that the information in the identifier is identical to the first file we inspected, with the exception that there is a `/2` at the end. This indicates that these reads are the second set in what are known as *paired-end* reads, as were introduced in the above video. The two files will have this identical structure where the order of the sequences in one is identical to the order of the sequences in the other. This way when they are read as a pair of files, they can be stepped through read-by-read & the integrity of the data will be kept intact.

The Illumina Chastity Filter

It is also worth noting that the reads we've just glanced at come from a version of the Illumina *casava* pipeline which is <1.8, and which is a relatively common format. The casava version really just describes how old the software on the Illumina machine is & we don't choose this. For more recently generated reads where the casava software is >1.8 of the casava, there is an additional field in the identifier which indicates whether a read would have *failed* an initial QC check. An example of this format would be:

```
@D5B4KKQ1:554:C4YHPACXX:4:1101:1084:2100 1:Y:0:
```

Note the "Y" in the final fields, which indicates this sequence would have **failed** QC. These low-quality reads were automatically removed in earlier versions of the pipeline and were omitted from the fastq file. However, they are now included with this additional field indicated in the read identifier. Inspection of this line will enable you to find out which version of the casava pipeline has been used, and whether you need to perform any additional filtering steps to remove low quality reads. The tool `fastq_illumina_filter` is designed to remove these reads for you & the tool, along with usage instructions can be

found at http://cancan.cshl.edu/labmembers/gordon/fastq_illumina_filter/.

An alternate set of reads which we'll also explore today is in included in the directory `~/NGSData/iCLIP/rawData`. These are immunopreciptated mRNAs and were supplied by some collaborators who'd either forgotten the removal of low-quality reads or had deliberately neglected step to increase read numbers. Note that these files are not compressed and are essentially plain text files, so we can directly inspect them using the command `head` instead of decompressing first.

```
cd ~/NGSData/iCLIP/rawData
head -n12 iCLIP_Sample1.fastq
```

Note that the header lines follow the format described on the previous page with this additional field at the end of the line.

Of the first three reads which have been printed to the terminal above, which one(s) will have failed the chastity filter, and which ones will have passed the filter?

Quality Scores

The only other line in the fastq format that really needs some introduction is the quality score information. These are presented as single *ascii* text characters for simple visual alignment with the sequence, and each character corresponds to a numeric value, which is the quality score. In the ascii text system, each character has a numeric value which we can interpret as an integer. Head to the website with a description of these at http://en.wikipedia.org/wiki/ASCII#ASCII_printable_code_chart.

The first 31 characters are non-printable & contain things like end-of-line marks and tab spacings, and note that the first printable character after the space is '!' which corresponds to the value 33. In short, the values 33-47 are symbols like !, ; #, $ etc, whereas the values 48-57 are the characters 0-9. Next are some more symbols (including @ for the value 64), with the upper case characters representing the values 65-90 & the lower case letters representing the values 97-122.

In the line of quality scores for the first read in the iCLIP sample above, the first character is "=", so this base has a quality score of 61. The second symbol in this line is"D", so the second base has a quality score of 68.

The PHRED +33/64 Scoring System

Now that we understand how to turn the quality scores from an ascii character into a numeric value, we need to know what these numbers represent. The two main systems in common usage are PHRED +33 and PHRED +64 and for each of these coding systems we either subtract 33 or 64 from the numeric value associated with each ascii character to give us a PHRED score. As will be discussed later, this score ranges between 0 and about 41.

The PHRED system used is determined by the software installed on the sequencing machine, with early machines using PHRED + 64 (casava <1.5), and more recent machines tending to use PHRED + 33. For example, in PHRED +33, the @ symbol corresponds to Q = 64 - 33 = 31, whereas in PHRED +64 it corresponds to Q = 64 - 64 = 0.

The following table demonstrates the comparative coding scale for the different formats:

```
..............................IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII.....................
.................................JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ.....................
LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL.....................................................
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
|                          |    |        |                              |                     |
33                         59   64       73                             104                   126

I - Illumina 1.3+ Phred+64, raw reads typically (0, 40)
J - Illumina 1.5+ Phred+64, raw reads typically (3, 40)
L - Illumina 1.8+ Phred+33, raw reads typically (0, 41)
```

Have a look at the quality scores in the RNA-Seq data. Which coding system do you think has been used for the RNA-Seq reads that we have?
In the PHRED +33 coding system, the character '@' is used. Can you think of any potential issues this would cause when searching within a fastq file? (Hint: Consider the sequence identifier rows)

Interpretation of PHRED Scores

All NGS platforms have non-zero error rates during the sequencing process (http://www.molecularecologist.com/next-gen-table-3c-2014/), and the quality scores are related to the probability of calling an incorrect base. The PHRED scoring system predates NGS technologies & is based on whether the wavelength from one specific base is clearly dominant through the formula

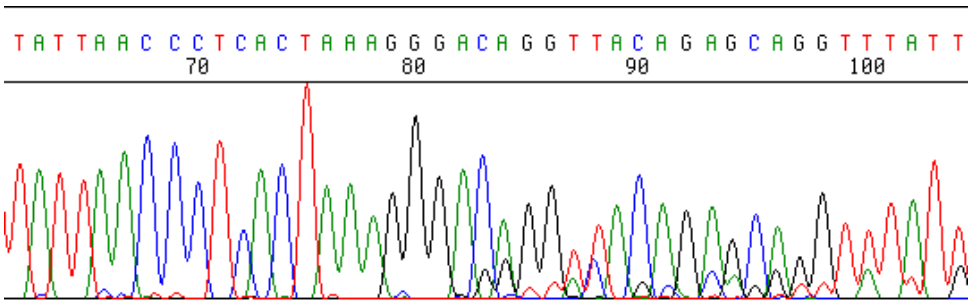
$$Q = -10log_{10}P \tag{1}$$

where P is the probability of calling the incorrect base.

This is more easily seen in the following table:

PHRED Score	Probability of Incorrect Base Call	Accuracy of Base Call
0	1 in 1	0%
10	1 in 10	90%
20	1 in 100	99%
30	1 in 1000	99.9%
40	1 in 10000	99.99%

As each base is added, the light wavelength associated with each base (i.e. the colour) is detected. In theory, there should only be one colour observed, but in reality there will always be a residual signal from the other bases. The following chromatogram based on the traditional Sanger sequencing methods demonstrates this well (Source: http://seqcore.brcf.med.umich.edu.



Note that early in this sequence, the colour peaks are clear but with small residual peaks of background signal. These peaks would receive high PHRED scores. Around position 83 however, the peaks become less obvious and any corresponding PHRED score would be lower. This basic principle is applied on a massively parallel scale during the generation of NGS data.

A common threshold for inclusion of a sequence is that all bases must have a Q score > 20. Considering the millions of sequences obtained from a flowcell, do you think that NGS data is likely to be highly accurate?

FASTQC

Primary Author(s):
Steve Pederson, Bioinformatics Hub, University of Adelaide
stephen.pederson@adelaide.edu.au

Contributor(s):
Dan Kortschak, Adelson Research Group, University of Adelaide
dan.kortschak@adelaide.edu.au
Terry Bertozzi, SA Museum Terry.Bertozzi@samuseum.sa.gov.au

Using fastqc

Removal of low-confidence base calls is an important part of any NGS analysis, and we can begin this process by checking the quality of libraries using the tool `fastqc`. As with all programs on the command line, we need to see how it works before we use it. The following command will open the help file in the `less` pager which we used earlier. To navigate through the file, use the <spacebar> to move forward a page, <b> to move back a page & <q> to exit the manual.

```
1 fastqc -h | less
```

Fastqc will create an html report on each file you submit, which can be opened from any web browser, such as `firefox` As seen in the help page, `fastqc` can be run from the command line or from a graphic user interface (GUI). Using a GUI is generally intuitive, so today we will look at the command line usage, as that will give you more flexibility & options going forward. Some important options for the command can be seen in the manual.

As you will see in the manual, setting the `-h` option as above will call the help page. Look up the following options to find what they mean.

Option	Usage
-o	
-t	
--casava	

As we have two RNA-Seq files, we will first need to create the output directory, then we can run `fastqc` using 2 threads which will ensure the files are processed in parallel. This can be much quicker when dealing with large experiments.

```
1 cd ~/NGSData/RNASeq/rawData
2 mkdir -p ~/QC/RNASeq/rawData
3 fastqc -o ~/QC/RNASeq/rawData -t 2 reads1.fq.gz reads2.fq.gz
```

It's probably a good idea to scribble a note next to each line if you didn't understand what you did. If you haven't seen the command `mkdir` before, check the help page

```
1 man mkdir
```

The above command gave both files to fastqc, told it where to write the output (`-o ~/QC`) & requested two threads (`-t 2`). The reports are in the html files, with all of the plots stored in the zip files. To look at the QC report for each file, we can use `firefox`.

```
cd ~/QC/RNASeq/rawData
ls -lh
firefox reads1.fq_fastqc.html reads2.fq_fastqc.html &
```

The left hand menu contains a series of click-able links to navigate through the report, with a quick guideline about each section given as a tick, cross of exclamation mark.

Two hints which may make your inspection of these files easier are:

1. To zoom out in `firefox` use the shortcut Ctrl-. Reset using Ctrl0 and zoom in using Ctrl+
2. You can open these directly from a traditional directory view by double clicking on the .html file.

If your terminal seems busy after you close `firefox`, use the 'Ctrl C' shortcut to stop whatever is keeping it busy

How many reads are there in both files?
How long are the reads in these files?

Interpreting the FASTQC Report

As we work through the QC reports we will develop a series of criteria for filtering and cleaning up our files. There is usually no perfect solution, we just have to make the best decisions we can based on the information we have. Some sections will prove more informative than others, and some will only be helpful if we are drilling deeply into our data. Firstly we'll just look at a selection of the plots. We'll investigate some of the others with some 'bad' data later.

Per Base Sequence Quality

Both of the files should be open in `firefox` in separate tabs. Perform the following steps on both files. Click on the `Per base sequence quality` hyperlink on the left of the page & you will see a boxplot of the QC score distributions for every position in the read. This is the main plot that researchers will look at for making informed decisions about later stages of the analysis.

What do you notice about the QC scores as you progress through the read?

We will deal with trimming the reads in a later section, but start to think about what you should do to these reads to ensure the highest quality in your final alignment & analysis.

Per Tile Sequence Quality This section just gives a quick visualisation about any physical effects on sequence quality due to the tile within the each flowcell. For the first file, you will notice an even breakdown in the quality of sequences near the end of the reads across all tiles. In our second QC report, you will notice a poor quality around the 25th base in the 2nd (or 3rd) tile. Generally, this would only be of note if drilling deeply to remove data from tiles with notable problems. Most of the time we don't factor in spatial effects, unless alternative approaches fail to address the issues we are dealing with.

Per Sequence Quality Scores This is just the distribution of average quality scores for each sequence. There's not much of note for us to see here.

Per Base Sequence Content This will often show artefacts from barcode sequences or adapters early in the reads, before stabilising to show a relatively even distribution of the bases.

Sequence Duplication Levels This plot shows about what you'd expect from an RNA-Seq experiment. There are a few duplicated sequences (rRNA?, highly expressed genes? etc.) and lots of unique sequences represented the diverse transcriptome. This is only calculated on a small sample of the library for computational efficiency and is just to give a rough guide if anything unusual stands out

Kmer Content Statistically over-represented sequences can be seen here & often they will overlap. In our first plot, the some sequences derive from the same motif, and are the extracts of the same longer sequence, just shifted along one base. No information is given as the source of these sequences, and you would expect to see barcode sequences or motifs that correspond to any digestion protocols here. This is a plot that can cause significant confusion, but can alert you to any unexpected sequence-based problems in your data.

Some Flawed Data

This RNA-Seq dataset is relatively "well-behaved", so let's look at another dataset which is a bit more problematic. First we'll need to create the directory structure and download the files.

```
cd
mkdir -p NGSData/iCLIP/rawData
cd NGSData/iCLIP
wget \
   https://swift.rc.nectar.org.au:8888/v1/AUTH_33065ff5c34a4652aa2fefb292b3195a/IntroNGS
cd rawData
wget \
   https://swift.rc.nectar.org.au:8888/v1/AUTH_33065ff5c34a4652aa2fefb292b3195a/IntroNGS
unzip iCLIP.zip && rm iCLIP.zip
```

Let's run `fastqc` & inspect the iCLIP dataset, which is a version of RNA-Seq data where the RNA has been pulled down via IP after cross-linking to an RNA-associated protein (Argonaute).

```
mkdir -p ~/QC/iCLIP/rawData
cd ~/NGSData/iCLIP/rawData
fastqc -o ~/QC/iCLIP/rawData -t 2 iCLIP_Sample1.fastq iCLIP_Sample2.fastq
cd ~/QC/iCLIP/rawData
firefox iCLIP_Sample1_fastqc.html iCLIP_Sample2_fastqc.html &
```

There are several things to note about these reports. Firstly, we know from inspection that these fastq libraries contain reads which have failed the Illumina Chastity filter. We would expect these to show up in the *Basic Statistics* table, but they don't by default.

How would we know that there are low quality reads here if the Illumina flag is ignored?

To see how these reads look without the flagged reads, we can turn on the `--casava` option when running `fastqc`. You can ignore any warning messages you receive about not looking like part of a CASAVA group.

```
cd ~/NGSData/iCLIP/rawData
fastqc --casava -o ~/QC/iCLIP/rawData -t 2 iCLIP_Sample1.fastq \
    iCLIP_Sample2.fastq
cd ~/QC/iCLIP/rawData
firefox iCLIP_Sample1_fastqc.html iCLIP_Sample2_fastqc.html &
```

Note that the above code will have over-written the original QC reports. As this second analysis is the best approach, this is actually OK at this point. However. this can be a common pitfall when running an analysis, which is why we will often record our analysis as a script. That way we have a record of what we have done and know exactly what options we have set for a process.

Adapter Contamination Go to the over-represented sequences section for Sample1. There seem to be a large number of over-represented sequences here which are not listed as matching any known sequences. Scanning through these manually is near impossible and solving this may take some leg work. Shift over to the same section in Sample2 and you will notice about 23% of sequences seem to contain an Illumina Paired End PCR Primer. The next line also contains matches to this but with some indels.

Now head to the *Adapter Content* section of each report & consider what we are seeing here. By the time you reach the 60th nucleotide in the reads from Sample1, this plot tells you that $> 50\%$ of reads contain matches to the Illumina Universal Adapter.

Some More Example Reports

We'll actually try to clean the iCLIP dataset up in the next chapter, but for now let's just head to another sample plot at http://www.bioinformatics.babraham.ac.uk/projects/fastqc/bad_sequence_fastqc.html

Per Base Sequence Quality Looking at the first plot, we can clearly see this data is not as high quality as the one we have been exploring ourselves.

Consider that the minimum sequence length required for confidently obtaining unique alignments is ¿20bp. Two approaches to this data might be to only include high quality sequences, or to trim the low quality bases from the ends and use shorter reads for downstream analysis. What would be the consequences of either approach?

Per Tile Sequence Quality Some physical artefacts are visible & some tiles seem to be consistently lower quality. Whichever approach we take to cleaning the data will more than likely account for any of these artefacts. Sometimes it's just helpful to know where a problem has arisen.

Overrepresented Sequences Head to this section of the report & scan down the list. Unlike our sample data, there seem to be a lot of enriched sequences of unknown origin. There is one hit to an Illumina adaptor sequence, so we know at least one of the contaminants in the data. Note that some of these sequences are the same as others on the list, just shifted one or two base pairs. A possible source of this may have been non random fragmentation.

Kmer Content

Do you notice anything unusual about this plot?

Interpreting the various sections of the report can take time & experience. A description of each of the sections is available from the `fastqc` authors at http://www.bioinformatics.babraham.ac.uk/projects/fastqc/Help/

Another interesting report is available at http://www.bioinformatics.babraham.ac.uk/projects/fastqc/RNA-Seq_fastqc.html Whilst the quality scores generally look pretty good for this one, see if you can find a point of interest in this data. This is a good example, of why just skimming the first plot may not be such a good idea.

In our dataset of two samples it is quite easy to think about the whole experiment & assess the overall quality. What about if we had 100 samples? Each .zip archive contains text files with the information which can easily be parsed into an overall summary.

Whilst this will require low-level scripting skills to perform on an experiment, we can quickly look at two of the important files today. The overall summary in terms of PASS/FAIL is contained in the 'summary.txt' file within the archive. Open this file in the `less` pager, and once you've had a look type `q` to quit, as we have become familiar with. First make sure you are in the correct directory, then inspect these files using `less`.

```
cd ~/QC/RNASeq/rawData
unzip -oc reads1.fq_fastqc.zip '*summary.txt' | less
```

The raw numbers for each of the sections are in the file fastqc_data.txt. Page through the file, until you lose interest then quit the pager.

```
unzip -oc reads1.fq_fastqc.zip '*fastqc_data.txt' | less
```

We can also extract any specific image file for compiling into a pdf, or find whatever we need by using these ideas. This makes handling the data for a large experiment

much simpler. There are plenty of hints online for how to write a *shell script*, or alternatively, attend one of our scripting workshops.

Further Reading

An excellent article which deals with some of the issues around data quality is:

Zhou, X and Rokas, A. (2014). *Prevention, diagnosis and treatment of high-throughput sequencing data pathologies.* Molecular Ecology 23, 1679-1700.

This has been included on your VM as the file QC.pdf & contains many examples of good data and low quality data, as well as a detailed discussion. If you feel like you are running ahead of schedule, or if you finish early it may be a good opportunity to download & read through the article. The workflow given at the end may also be particularly useful.

Space for Personal Notes or Feedback