**Pipeline Week 1 – Notepad**

Docker machine is 10.182.155.27 and port 52024

login is stpuser - pass stpuser

From SOP: ‘Data is available inside the docker at /example\_fastqs and this points to /sdgs/teaching/example\_fastqs on the host. Any data generated by the trainee should be stored in /home/stpuser, this points to /sdgs/teaching/personal\_storage/<USER> on the host. This ensures any data they create is persistent in the event of a problem with their docker.’

**How to write up the final pipeline as a script and run it**

Make a new file in the Docker using WinSCP. Save it with the extension .sh

Write your pipeline code in this file.

Within the terminal, access the correct directory and type bash filename.sh

The script will execute.

**TASKS**

DONE Perform FASTQC on FASTQs for quality control

DONE Start to map the reads (align them to reference) – pick software for this

DONE Read best practice guidelines: ACGS, ACMG, GATK

**What are FASTQ files? What is the purpose of FASTQC?**

FASTQ files are generally the first file out of the machine, at least for Illumina sequencers

(Wikipedia) FASTA format plus an ASCII encoded quality score, 4 lines

No standard file extension but .fq and .fastq are common

**The fastq -> alignment step is covered by Pre-Processing in the GATK guidelines**

https://software.broadinstitute.org/gatk/best-practices/bp\_3step.php?case=GermShortWGS

The FASTQ files need to be quality checked.

Then mapped to a reference genome

Makes a SAM/BAM file sorted by coordinate.

Mark duplicates

Recalibrate base quality scores (to account for coverage instead of basing off a single sequence’s quality score)

**Performing the mapping to make a SAM/BAM file**

GATK Best Practice: recommends BWA MEM for DNA but will depend on data and how it was sequenced

Recommends checking that reads are properly sorted by coordinate afterwards

**Using FASTQC**

**Quality control** tool (not mapping) written in Java which uses the Picard BAM/SAM libraries

Information on running it at: https://www.bioinformatics.babraham.ac.uk/projects/fastqc/INSTALL.txt

Linux: We have included a wrapper script, called 'fastqc' which is the easiest way to

start the program. The wrapper is in the top level of the FastQC installation. You

may need to make this file executable:

chmod 755 fastqc

..but once you have done that you can run it directly

./fastqc

..or place a link in /usr/local/bin to be able to run the program from any location:

sudo ln -s /path/to/FastQC/fastqc /usr/local/bin/fastqc

**How to run fastqc?**

To run non-interactively you type the command then specify a list of files to process on the commandline:

./fastqc somefile.txt someotherfile.txt

**How would I run it on all the files in a directory?**

May be possible using the second (non-for-loop) answer here: https://stackoverflow.com/questions/10523415/bash-script-to-execute-command-on-all-files-in-a-directory

**TEST WITH TWO FILES (1504850-S1509352-02\_GCTCGGTA\_L001\_R1\_001.fastq.gz 1607686-S1615531-02\_TTCACGCA\_L002\_R2\_001.fastq.gz)**

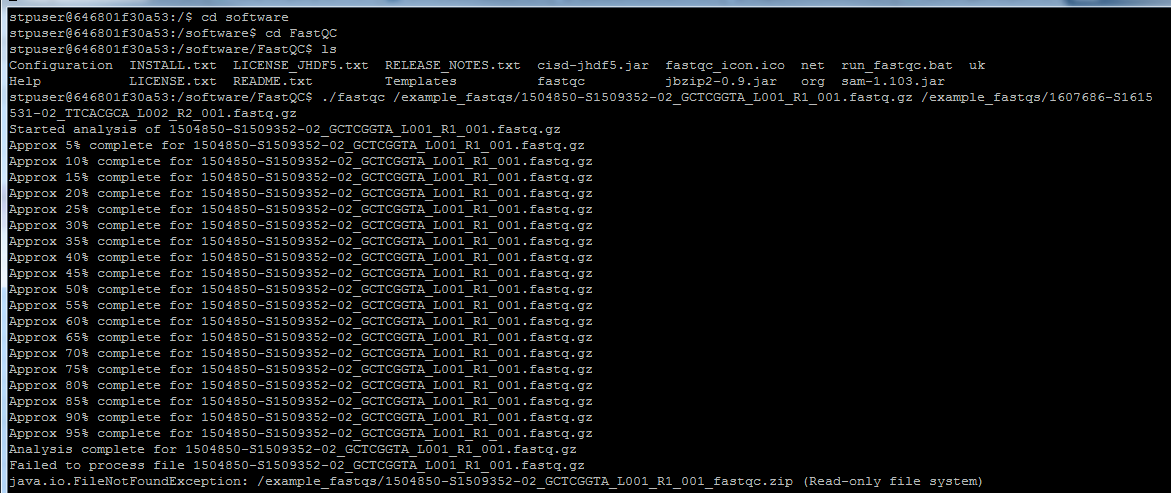
‘Didn’t exist or couldn’t be read’ – need to do file paths probably? So with /example\_fastqs/ in front of each identifier.

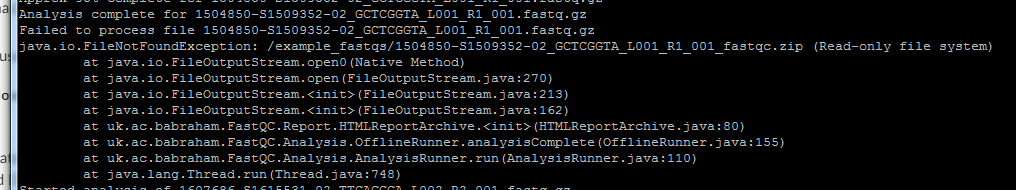
./fastqc 1504850-S1509352-02\_GCTCGGTA\_L001\_R1\_001.fastq.gz 1607686-S1615531-02\_TTCACGCA\_L002\_R2\_001.fastq.gz

It ran! But I don’t know where the result went. Says it automatically outputs to the location of the fastqs. So, should be in /example\_fastqs/.

I don’t see it ☹

Failed as ‘read-only filesystem’? It completed analysis but failed to complete processing – seems to be issues with the output stream – therefore it may be because of not being able to write to the default example\_fastqs folder.





Spoke to Lizzy: I didn’t have permission to write to home/stpuser but Matt has granted that now.

Try running test sample again but write to home/stpuser this time, which you do with:

./fastqc 1504850-S1509352-02\_GCTCGGTA\_L001\_R1\_001.fastq.gz 1607686-S1615531-02\_TTCACGCA\_L002\_R2\_001.fastq.gz --outdir=/home/stpuser

Ran this time without errors. Need to check that it went to the stpuser directory

It did!

**Run FASTQC on all of the example\_fastqs and output them to home/stpuser/second\_run**

For just our two test samples, the command made from the software/fastqc directory would be:

./fastqc 1504850-S1509352-02\_GCTCGGTA\_L001\_R1\_001.fastq.gz 1607686-S1615531-02\_TTCACGCA\_L002\_R2\_001.fastq.gz --outdir=/home/stpuser/second\_run

This is how to feed all of the fastq files from example\_fastqs into the ./fastqc command without typing all their stupid names out:

../../example\_fastqs/\*fastq.gz

So the total command from within the FastQC directory is:

./fastqc ../../example\_fastqs/\*fastq.qz --outdir=/home/stpuser/second\_run

Done! 20 html files retrieved

I can open some html files from WinSCP, so I don’t have to worry about downloading things to let me open files from bash (see example below).

**How is quality looking for some of these examples?**

Quality seems mostly fine:

1504850-S1509352-02\_GCTCGGTA\_L001\_R1\_001\_fastqc

Slight issues:

1504850-S1509352-02\_GCTCGGTA\_L001\_R2\_001\_fastqc

Sequence quality, number of Ns high over bases 95-100

1504850-S1509352-02\_GCTCGGTA\_L002\_R1\_001\_fastqc

Bad quality tile ~32, sequence content drops off at 96-7

1504850-S1509352-02\_GCTCGGTA\_L002\_R2\_001\_fastqc

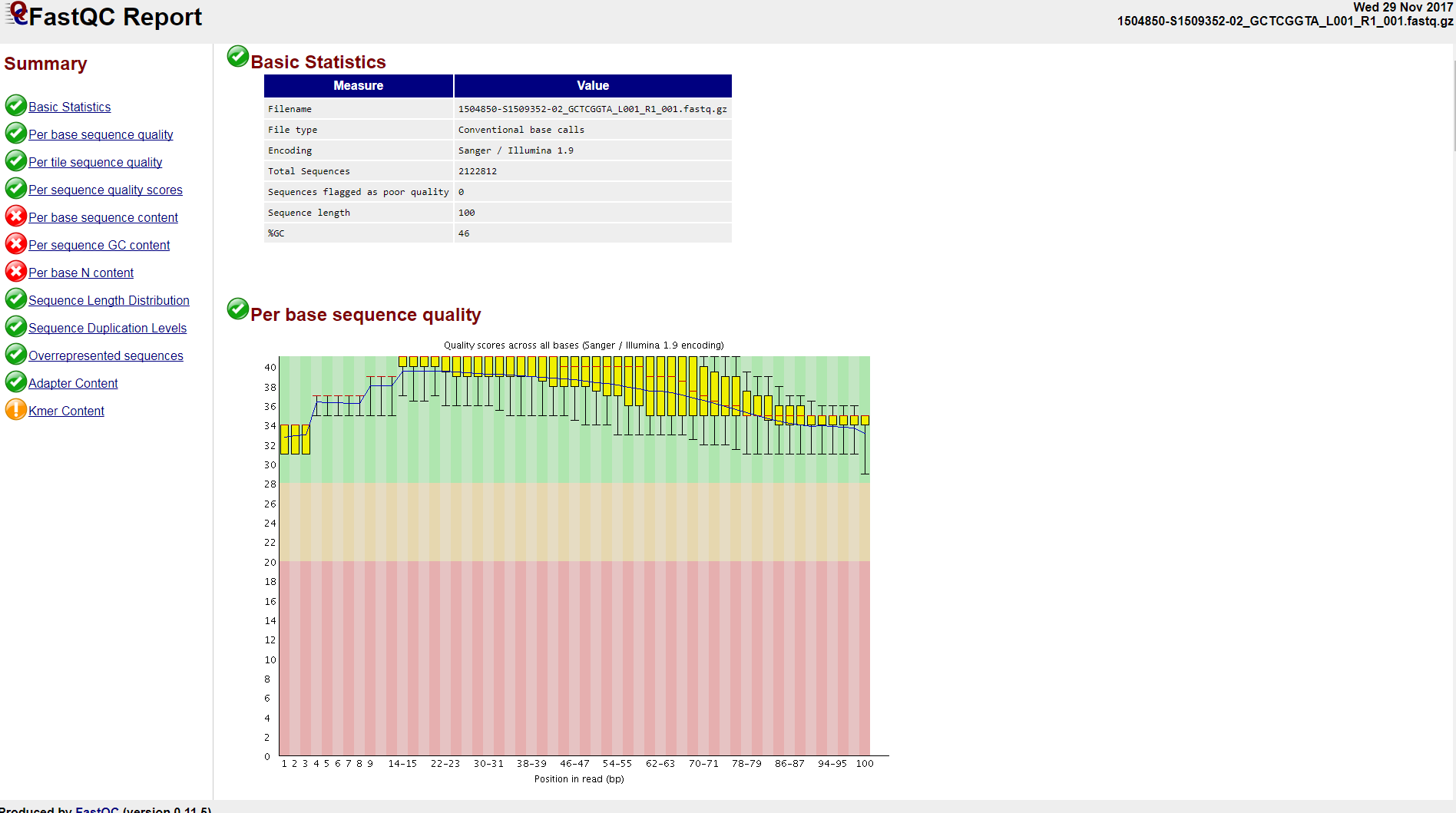
Again, drop-offs and Ns above 96-7

1606034-S1612259-02\_CGAACTTA\_L001\_R1\_001\_fastqc

Kmer content flags, some metrics dropping towards sequence end.

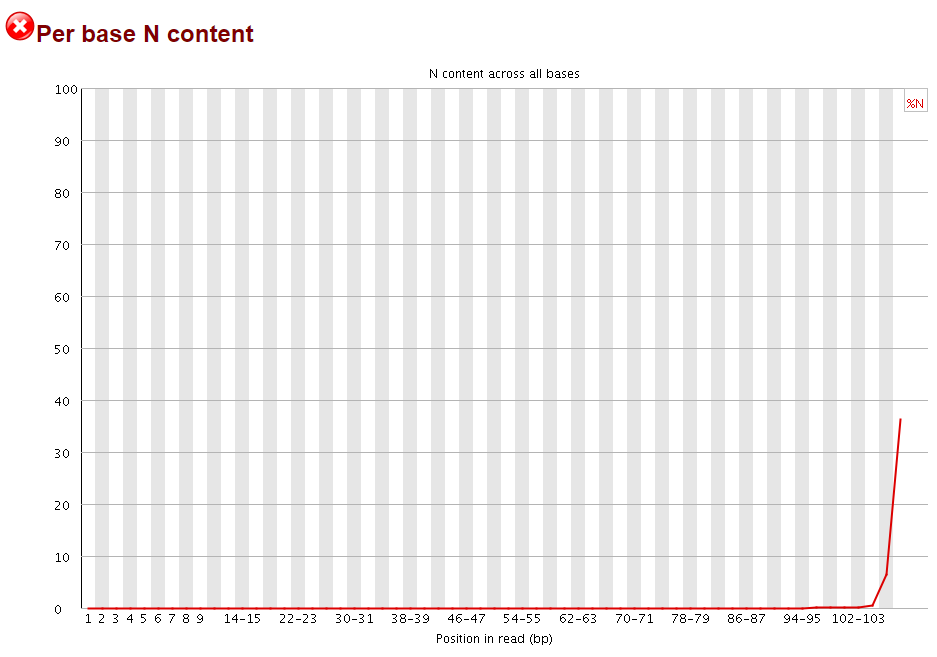
Particularly bad or nonsense (glanced over every sample and compared to the ‘slight issues’ category):

None, thankfully



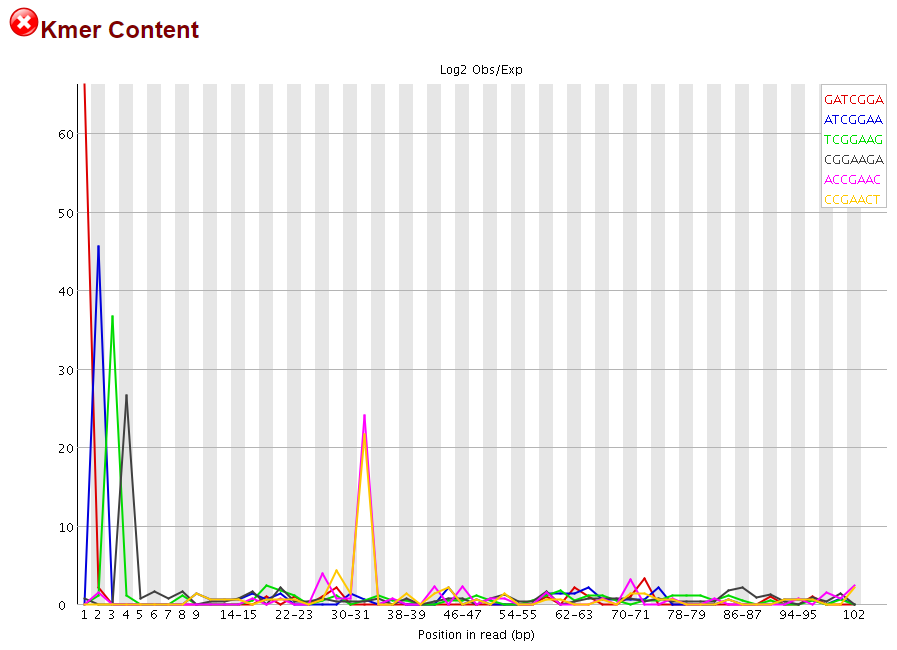
Picture: Example html report for 1504850-S1509352-02\_GCTCGGTA\_L001\_R1\_001.fastq.gz.

First 6 bases or so – calibrating – lower quality sometimes results.



Pictured: An example of an N content increase at higher base numbers. Other metrics such as sequence quality also tend to decline towards the end of the sequence.

Ns at the end can be due to ADAPTOR READTHROUGH, which are then trimmed by replacing with Ns (‘soft clipped’).



Pictured: An example of a warning-flagged kmer content.

**Mapping reads against a reference**

Which software to use?

GATK likes BWA MEM and this is available

Software/bwa-0.7.15/bwa brings up a help page, which explains that typing bwa mem should call the BWA MEM algorithm

It also says that you need to first index the genome with ‘bwa index’

Which reference to use?

Relevant directory is within root

/reference\_files

Available references are:

stpuser@646801f30a53:/reference\_files$ ls

1000G exac ucsc.hg19.nohap.masked.fasta.ann

annovar gatk\_resource ucsc.hg19.nohap.masked.fasta.bwt

clinvar ucsc.hg19.nohap.masked.dict ucsc.hg19.nohap.masked.fasta.fai

cosmic ucsc.hg19.nohap.masked.fasta ucsc.hg19.nohap.masked.fasta.pac

dbsnp ucsc.hg19.nohap.masked.fasta.amb ucsc.hg19.nohap.masked.fasta.sa

Most of these references (exac e.t.c.) are more relevant to annotation than to actually aligning

We want ucgs.hg19.nohap.masked.fasta

Potential handy resource: https://hcc-docs.unl.edu/display/HCCDOC/Running+BWA+Commands

Check your permissions! (Says Christine)

ls –l

Returns a code you’ll need to look up

Make index of the reference genome

The example given online is:

bwa index [-a bwtsw|is] input\_reference.fasta index\_prefix

Note that –a bwtsw is an obligatory option which does not work for short genomes (the other, ‘is’, does not work for long genomes).

My attempt is:

bwa index -a bwtsw ../../reference\_files/ ucsc.hg19.nohap.masked.fasta test\_prefix

It doesn’t like it; returns errors saying the reference genomes is in a read-only file system

The good news: I can skip the index part as it seems to just use the one available already

Example code:

bwa mem index\_prefix [input\_reads.fastq|input\_reads\_pair\_1.fastq input\_reads\_pair\_2.fastq]

My attempt (ucsc.hg19.nohap.masked.fasta.fai is used here because .fai is a suffix for the index of a fasta file):

bwa mem ../../reference\_files/ucsc.hg19.nohap.masked.fasta.fai ../../example\_fastqs/\*fastq.gz

Just returns a list of algorithm options, suggesting it didn’t run. Also didn’t work when I tried again but put ./ in front of bwa mem.

I need to output it to its own file!

--outdir=/home/stpuser flags as an invalid option

It’ll have its own stupid syntax probably

Someone online suggested (index) (files to map) > (output)

stpuser@646801f30a53:/software/bwa-0.7.15$ ./bwa mem ../../reference\_files/ucsc.hg19.nohap.masked.fasta.fai ../../example\_fastqs/\*fastq.gz > ../../home/stpuser -bash: ../../home/stpuser: Is a directory

CORRECT FORMAT: > ../../home/stpuser/**aln.bam**

It writes the sequence file for you – no need to make one in advance

But it SHOULD output to the screen and not fail out

Is it because I’m trying to run multiple fastq.gz files without looping them?

Almost definitely

I need the SourceForge advice pages but they are blocked

What is ${} in bash?

From Stack Exchange:

‘By contrast, ${variable} is just a disambiguation mechanism, so you can say ${var}text when you mean, the contents of the variable var, followed by text (as opposed to $vartext which means, the contents of the variable vartext).’

Looking at the real pipeline – line 543 is the FASTQ alignment bit

Each sample is going to have 4 SEQUENCES – two forwards which are duplicates of one another, two reverses which are duplicates of one another

This is a HiSeq thing

Try with just one sequence to start?

1504850-S1509352-02\_GCTCGGTA\_L001\_R1\_001.fastq.gz

./bwa mem ../../reference\_files/ucsc.hg19.nohap.masked.fasta.fai ../../example\_fastqs/1504850-S1509352-02\_GCTCGGTA\_L001\_R1\_001.fastq.gz > ../../home/stpuser/**aln.bam**

Returns error: ‘fail to locate the index files’

Ran again with just .fasta reference, not the .fai index (wants to find it itself I guess?)

It ran!

How do I combine the 4 sequences?

Mem – might be able to concatenate the 2 files of each (2 forward, 2 reverse) and run it with these two combined

An example of the files:

Lane number, read number

1703057-S1705957-02\_AAGACGGA\_L001\_R1\_001.fastq.gz

1703057-S1705957-02\_AAGACGGA\_L001\_R2\_001.fastq.gz

1703057-S1705957-02\_AAGACGGA\_L002\_R1\_001.fastq.gz

1703057-S1705957-02\_AAGACGGA\_L002\_R2\_001.fastq.gz

Lanes are like repeats, reads are ‘side-by-side’ on the genome (and may have an overlap depending on insert size)

Can merge lanes like:

 cat \*R1\_.fastq > both\_R1.fastq

This makes a file with one line below the other, essentially

My version for if we **only** had a single pair (not general!) would be:

cat \*R1\_001.fastq.gz > twolanes\_R1\_001.fastq

bwa mem is perfectly happy to use **read 1 and read 2** at the same time, so we don’t need to glue reads 1 and 2 together in advance of running it:

bwa mem -M -t 16 ref.fa read1.fq read2.fq > aln.sam

How do I loop through the concatenation?

$(x) performs command x (which can be a command such as ls) and puts it in its place

For loop sample:

for i in $( ls ); do

echo item: $i

done

Stack Exchange example for merging **4** lanes (R1s only):

for name in ./\*.fastq.gz; do

rnum=${name##\*\_}

rnum=${rnum%%.\*}

sample=${name#\*\_}

sample=${sample%%\_\*}

cat "$name" >>"${sample}\_$rnum.fastq.gz"

done

Cheap version: run this twice, once for R1.gz and once for R2.gz?

for name in ../../example\_fastqs/./$\*\_R1.fastq.gz; do

rnum=${name##\*\_}

rnum=${rnum%%.\*}

sample=${name#\*\_}

sample=${sample%%\_\*}

cat "$name" >> ../../home/stpuser/"${sample}\_$rnum\_R1.fastq.gz"

done

for name in ../../example\_fastqs/./\*\_R2.fastq.gz; do

rnum=${name##\*\_}

rnum=${rnum%%.\*}

sample=${name#\*\_}

sample=${sample%%\_\*}

cat "$name" >> ../../home/stpuser/"${sample}\_$rnum\_R2.fastq.gz"

done

Tried the first one: ‘/home/stpuser/fastqs’ no such directory

<http://tldp.org/LDP/Bash-Beginners-Guide/html/sect_09_01.html>

This is nice and clear: <https://ryanstutorials.net/bash-scripting-tutorial/bash-loops.php>

Using awk for pattern matching

I know that I want to go through the subset of my example fastqs which are L001\_R1\_001, find their identically-named-except-for-R2 L001\_R2\_001 counterparts, and concatenate the second to the first. I don’t know the function for the second bit though. Christine says that awk might be good for the pattern-matching.

To print the filename with awk you use:

Awk ‘{print FILENAME}’ /example\_fastqs/\*fastq.gz

I can’t find any guides for how to pattern-name with this information though.

How to loop through and echo the fastq files

For I in /example\_fastqs/\*fastq.gz; do

echo ${i};

done

Awk: Split strings with a delimiter

Good little demonstration

<https://viewsby.wordpress.com/2012/09/14/awk-split-string-using-a-delimiter/>

#Following takes the file name, and pipes it to awk, which prints only the unique section before the first underscore

echo 1504850-S1509352-02\_GCTCGGTA\_L001\_R2\_001.fastq.gz | awk -F'\_' '{print $1}' 1504850-S1509352-02

Loop through and give the unique IDs

for i in \*fastq.gz; do

echo ${i}|awk -F'\_' '{print $1}';

done

We can also do this just to get unique IDs, lane and read

for i in \*fastq.gz; do

echo ${i}|awk -F'\_' '{print $1 “\_” $3 “\_” $4}';

done

Can feed these into variable x so it’s easier to handle later

for i in \*fastq.gz; do

x=$(echo ${i}|awk -F'\_' '{print $1 “\_” $3 “\_” $4}');

echo $x

done

Grep is like the bash equivalent of Ctrl+F

Grep x /etc/passwd

# returns words from /etc/passwd containing x

Grep –n x /etc/passwd

# returns line numbers containing x

Can we take each unique ID and look for others (doesn’t do anything with them yet)?

for i in \*fastq.gz; do

x=$(echo ${i}|awk -F'\_' '{print $1}');

echo $x

matches=$(grep $x \*fastq.gz)

echo $matches

done

It seems to be working! Outputs each unique identifier 4 times

6th December 2017

For now, just combine everything long-hand, as I’m falling behind.

Automate later?

SEE: cat\_lanes\_and\_align.sh

I managed to make two alignment files but the Brain server filled up. One file was truncated and so was deleted.

Converted the one Sam file to a bam file.

At Lizzy’s suggestion, I altered my script to pipe the bwa mem output into samtools, converting straight to a bam so we save space:

| samtools view -b -o

**Pipeline Week 2 – Notepad**

**TASKS**

DONE Finish alignments once there’s enough room

DONE Sort BAMs in order

DONE Mark duplicates

DONE Visualize sequence in IGV/Alamut

DONE Carry out indel realignment

Look at off-target reads

Start some coverage analysis

Do a QC report

**GATK Best Practice for pre-processing after bwa mem**

Sort reads by co-ordinate (?)

Mark duplicates because they aren’t informative

We use Picard

We also have sambamba: **sambamba-markdup**

Recalibrate base quality scores – this uses machine learning

BQSR recommended; model of covariation based on data and set of known variants. Optional – can build a second model and make before/after plots

**We don’t currently do this** but I can try if I like

**NOT IN GATK BUT IN OURS** Checking off-target reads

‘Intersect bed’ in bedtools

**Tuesday late afternoon lecture/discussion**

**Fun facts from Lizzy:**

‘Zcat’ can read zipped files for you (including GNU zipped files ending in .gz)

‘Gunzip’ will unzip your file and save it without the .gz

Piping bash commands into ‘less’ will slow down the printed output on the console so you can read it as it goes

df –lh checks the remaining space (in this case on the brain)

**Analysing coverage**

Sambamba can carry this out – give it regions or it’ll try to look at coverage across the whole genome

Maybe filter x30?

Coverage - sambamba depth on the whole (master) bed, then see how this intersects

There are separate exonic bed files available inside:

Bioinfo -> results -> analysis -> miseq -> masterBED

The master bed is the exons +/- 25 bases

The exonic\_files are the exons +/- 5 bases

Exonic files have 30x coverage, while the non-exonic-file bases in the master beds have x18 coverage (because those bits are less clinically significant)

**Post-alignment base pair quality**

We don’t do much with the base pair quality – no hard clips for instance – so quality on fastqc tends to decrease.

However, this is captured and dealt with by the variant caller.

**Off-target reads**

Intersect the bed files

Off targets grep –v and pipe to line counter

**Visualising a BAM file in Alamut**

I managed to get one BAM file before the brain filled up.

Copy to pipeline diary folder and open in Alamut.

Currently it won’t let me – the option under ‘Applications’ is greyed out and Ctrl+B isn’t opening it either.

Lizzy: I probably need an index file after sorting it. Use ‘samtools sort’ and then index it using ‘samtools index’.

#See script file

Samtools sort –o ../../home/stpuser/aligned\_seqs/1504850-S1509352-02\_GCTCGGTA.bam > ../../home/stpuser/aligned\_seqs/1504850-S1509352-02\_GCTCGGTA\_sorted.bam

Generated just a 1k file

Samtools index ../../home/stpuser/aligned\_seqs/1504850-S1509352-02\_GCTCGGTA\_sorted.bam

‘fail to open file’

Tried to rerun from script and it just says that it isn’t in a format that can be usefully indexed.

I want to try again when there’s more room on the server. I won’t be able to tell whether it’s a lack of room or just incorrect syntax until that is sorted out.

**How do I find the broad panel and small panel information for each patient?**

We need to find patients run on pipeline. In the sequence ID, the first number is WORKLIST and the second number is SAMPLE (which links to patient in Starlims).

You find the sequence data on the bioinfo server:

results -> Analysis -> HiSeq -> (year) -> (worklist) -> (sample) -> DevDel or analysis log

Broad-panel with be everything captured by target capture kit

If it’s OUTSIDE the broad-panel it’s an OFF TARGET READ

We don’t analyse the whole broad-panel, to reduce incidental findings

Small-panel will be the targeted regions of interest

This is also called a ‘virtual panel’

We do variant analysis on the small panel

A ‘reanalysis’ is when the small panel comes up blank, so we pick a new small panel from the original broad panel to search for variants in

The bed files are kept in:

results -> Analysis -> MiSeq -> MasterBED

**Sorting files**

Got syntax wrong initially; now able to sort with:

samtools sort -o ../../home/stpuser/aligned\_seqs/1504850-S1509352-02\_GCTCGGTA\_sorted.bam ../../home/stpuser/aligned\_seqs/1504850-S1509352-02\_GCTCGGTA.bam

Running for the remaining 4 samples

File name form is (worklist)-(sample)-(sample)-(index?)\_sorted.bam

**Viewing files**

Can do in IGV (click the .bat file)

But can’t do it by URL - must copy onto computer instead

Issue – don’t know where the bam has aligned so can’t find it on IGV (it’s miniscule compared to the whole genome); must find original sample bed, as when trying to look at off-target reads

1606034-S1612259-02

Broad-panel BED file CTDFinaldesignwith25bp\_v3.bed

small-panel BED file CTD\_EDS\_C\_25\_v4.bed

exonic small-panel BED file CTD\_EDS\_C\_25\_v4\_exonic.bed

EDS is probably the PanelApp Ehlers-Danlos panel – the full panel is A LOT of green genes

BED files: it’s classical EDS. So I need only look at chr2 COL5A2, chr9 COL5A1, chr17 COL1A1

**Marking duplicates (try Picard)**

Run from within Picard directory. I \*think\* metrics will be generated during the run and go into the .txt I wrote down.

cd ../picard-tools-2.5.0

java -jar picard.jar MarkDuplicates I=../../home/stpuser/aligned\_seqs/1504850-S1509352-02\_GCTCGGTA\_sorted.bam \ O=../../home/stpuser/aligned\_seqs/1504850-S1509352-02\_GCTCGGTA\_sorted\_dupflag.bam M= O=../../home/stpuser/aligned\_seqs/1504850-S1509352-02\_GCTCGGTA\_sorted\_dupflag\_metrics.txt

^This works fine – did for all the other files too

**Which bed files do we need for each sample?**

To work out reads off-target, I need to know the broad bed for the sample

Make a table:

|  |  |  |  |
| --- | --- | --- | --- |
| Sample | Broad panel bed | Small panel bed | Small panel exonic |
| 1504850-S1509352-02 | NGD\_CompletePanel\_25bp\_v2.bed | NGD\_dystonia\_v3\_25bp.bed |  |
| 1606034-S1612259-02 | CTDFinaldesignwith25bp\_v3.bed | CTD\_EDS\_C\_25\_v4.bed | CTD\_EDS\_C\_25\_v4\_exonic.bed |
| 1607686-S1615531-02 | Motor\_CompletePanel\_v1.bed | Motor\_HSP\_v2\_25bp.bed |  |
| 1609778-S1620040-02 | IEM\_all\_panels\_header.bed | IEM\_Rhabdo\_v3.bed |  |
| 1703057-S1705957-02 | HeredCancer\_full\_panel\_25bp\_v1.bed | HeredCancer\_BRCA1\_BRCA2\_only\_25\_v1.bed |  |

**Quality metrics – samtools flagstat**

**STARTED: using Git for this diary, and the pipeline shell script**

See how these compare against Picard?

Samtools flagstat

Did git commit for the samtools flagstat metrics

**Carry out indel realignment**

Indels tend to align poorly. This is because with repetitive regions, it’s hard to place individual sequences in the ‘right’ place along the reference. With indel realignment, you can look at the aggregated data of ALL the reads and use that to help make the cal. Generally indel realignment shoves things ‘leftward’ (for the forward strand).

GATK’s indel tool gives nice error messages (according to Christine) so I may try that first.

<https://gatkforums.broadinstitute.org/gatk/discussion/7156/howto-perform-local-realignment-around-indels>

First we will identify targets (problem areas) with RealignerTargetCreator. Then we actually carry out the realignment with IndelRealigner.

Tools will be in software/GenomeAnalysisTK-3.6

**Case of the missing read groups**

ERROR: RealignerTargetCreator fails because read group data isn’t in the BAM files. Tried to check header for @RG using grep but there isn’t a line which has this.

From <https://gatkforums.broadinstitute.org/gatk/discussion/6472/read-groups>: ‘In Illumina data, read group IDs are composed using the flowcell + lane name and number, making them a globally unique identifier across all sequencing data in the world.’

There are tools which will add read groups onto BAM files for you (Picard has AddOrReplaceReadGroups), if we can find out what the read groups were meant to be.

**Potential read group loss point: converting to BAM**

OR: maybe a header got left off in the sam -> bam piping step early on in the chain

Realign the first sequence and put that through the entire pipeline again to see what happens

Nope, -h at the SAM -> BAM conversion wasn’t the issue

It’s still missing from the BAM when we look at the header with grep

But the original fastq files have read group info, for sure

**Potential read group loss point: Bwa mem**

We need –R to keep full read group information

We also need to GIVE IT A STRING! –R *STR*

Currently trying to work out where we get that string from…

‘Deriving ID and PU fields from read groups’ <https://gatkforums.broadinstitute.org/gatk/discussion/6472/read-groups>

**The actual issue – we used cat on the Lane 1 and Lane 2 fastqs, and this probably destroyed read group information**

<https://gatkforums.broadinstitute.org/gatk/discussion/2730/merge-lanes-while-preserving-the-read-group-info>

It shouldn’t matter that we can for our samples – as we don’t run patient samples together, read group doesn’t have much relevance – but to be flexible in more general cases, I’d rather this was automatic and didn’t need re-adding.

New workflow to test on first sample:

1. Align each lane SEPARATELY in BWA mem (this is in GATK Best Practices)
   1. Write to temp files
2. Merge lane-level BAMs with Picard’s MergeSamFiles, or SamTools merge
3. Delete temp files
4. Continue as before with rest of pipeline

**Pipeline Week 4**

**Read groups aren’t automatically included in the Fastq**

I’ve changed the pipeline to follow best practice (merge not concatenate).

However I was focusing on preserving read group information.

Not the right approach – read group isn’t inherent to the fastq files!

Therefore, I still need to add read groups in using Picard

**Add read groups (Picard AddOrReplaceReadGroups)**

What should we call them?

java -jar picard.jar AddOrReplaceReadGroups \  
 I=input.bam \  
 O=output.bam \  
 RGID=4 \  
 RGLB=lib1 \  
 RGPL=illumina \  
 RGPU=unit1 \  
 RGSM=20

|  |  |
| --- | --- |
| RGID (String) | Read Group ID Default value: 1. This option can be set to 'null' to clear the default value. |
| RGLB (String) | Read Group library Required. |
| RGPL (String) | Read Group platform (e.g. illumina, solid) Required. |
| RGPU (String) | Read Group platform unit (eg. run barcode) Required. |
| RGSM (String) | Read Group sample name Required. |

RGID = 1

RGLB = lib1

RGPL = illumina

RGPU = unit1

RGSM = [sample name]

**Identify target regions**

Seems to have worked OK?

**Analysing coverage and off-target**

Picard seems to have straightforward-looking coverage tools: <https://broadinstitute.github.io/picard/picard-metric-definitions.html#HsMetrics>

Which beds do we have (so I know what depth to look for with each sample?):

|  |  |  |  |
| --- | --- | --- | --- |
| Sample | Broad panel bed | Small panel bed | Small panel exonic |
| 1504850-S1509352-02 | NGD\_CompletePanel\_25bp\_v2.bed | NGD\_dystonia\_v3\_25bp.bed |  |
| 1606034-S1612259-02 | CTDFinaldesignwith25bp\_v3.bed | CTD\_EDS\_C\_25\_v4.bed | CTD\_EDS\_C\_25\_v4\_exonic.bed |
| 1607686-S1615531-02 | Motor\_CompletePanel\_v1.bed | Motor\_HSP\_v2\_25bp.bed |  |
| 1609778-S1620040-02 | IEM\_all\_panels\_header.bed | IEM\_Rhabdo\_v3.bed |  |
| 1703057-S1705957-02 | HeredCancer\_full\_panel\_25bp\_v1.bed | HeredCancer\_BRCA1\_BRCA2\_only\_25\_v1.bed |  |

We have the small panel beds (+/- 25bp) and the broad beds.

We care about SMALL PANEL BED COVERAGE. These should be fully covered in each sample.

In our normal lab, the whole thing doesn’t need to be 30x. It only needs to be 30x for 5bp either side of the exon.

However I’m going to go for 30x so I don’t die of boredom.

USING: Picard HsMetrics

ALSO USING: Picard’s BedToIntervalList (because bed files need converting to interval files first)

**Decomposing, calling and annotating variants**

Variants called with GATK Haplotype Caller, constrained to SMALL bed (was broad but corrected it).

Multiallelic sites decomposed with Vt, though made no difference to the 15\* sample.

Variants annotated for human genome with SnpEff, constrained to the SMALL bed.

Attempted to annotate with dbSNP, but although it threw no warning messages, the rs numbers (or something similar) didn’t look like they were there. Might need fixing.

#Next: fix dbSNP annotation

#Filter out those with dbSNP frequency above 5%

**Making a log file**

Have inserted ‘echo’ lines to return start-and-end time, date and version of each item of software.

Need to make sure that it carries on OK if errors are thrown, and that these go to an error log.

**Rewriting bash script to take arguments**

Getting rid of hard-coding by feeding in arguments which are then assigned to variables inside the script (see conversation with Lizzy).