**Alignment post-processing**

Simon Rasmussen - June 2016

**Overview**

In this exercise we will pre-process bam-files so they are ready for SNP calling. This is necessary to reduce the high number of potential false SNPs that will get called. You will try to:

1. Sort and index BAM-files
2. Remove read duplicates from the BAM-files
3. Merge BAM files
4. Use GATK to perform realignment
5. Use GATK to perform Base Quality Recalibration (Optional)

**Preparation**

Go to your working folder and create a dir for this exercise

mkdir aln\_process

cd aln\_process

We are going to work on data from an asian individual (HG00418) that was sequenced to around ~40X using Illumina Paired end sequencing. We are not going to use all the data, but only some data from two libraries that are both paired end, and we are only going to use chr21 instead of the entire human genome. One of libraries are with an insert size of 180 and the other 481 and they are named "\_A" and "\_B" in the exercise below. If you did not finish the exercise or accidently deleted them you can copy the files from below, else you should copy them from your folder with the alignment exercise.

cp /home/27626/exercises/aln\_process/HG00418\_\* .

First try to take a look at the BAM-files:

samtools view -h HG00418\_A.bam | less -S

samtools view -h HG00418\_B.bam | less -S

Start by sorting the alignments using samtools. This short the alignments by chromosome position instead of the order they appeared in the fastq files.

samtools sort -O BAM -o HG00418\_A.sort.bam HG00418\_A.bam

samtools sort -O BAM -o HG00418\_B.sort.bam HG00418\_B.bam

Index the sorted alignments, this is needed for a lot of different applications.

samtools index HG00418\_A.sort.bam

samtools index HG00418\_B.sort.bam

**Merge bams to library level**

We will merge the alignments so that we have one file for each library that was sequenced. This would be the case if a library was sequenced on several lanes. **In our case we have one lane of each library so we do not need to merge**. In case you need to merge bam-files it can be done using samtools or picard. Example of picard command of merging lane1.bam and lane2.bam to libraryA.bam is shown here.

java -Xmx2g -jar /home/27626/bin/picard-tools-2.3.0/picard.jar MergeSamFiles INPUT=lane1.bam INPUT=lane2.bam OUTPUT=libraryA.bam

**Mark Duplicates**

Duplicates arise from artefacts during PCR amplification. What we will see in the alignment is that there will be many exact duplicates of a read. Because all the duplicated reads were sampled from the same DNA molecule it gives an uneven representation of that molecule compared to other molecules, and will bias the SNP calling. We will therefore remove them using Picard MarkDuplicates. Here duplicates are identified as reads that map with identical 5' coordinates and orientations, and only the reads with the highest quality score are kept for further analysis. We set REMOVE\_DUPLICATES=true so that duplicate reads are removed from the bam-file, else they will just be flagged as duplicates by adding 1024 to their flag value (column 2).

java -Xmx2g -jar /home/27626/bin/picard-tools-2.3.0/picard.jar MarkDuplicates INPUT=HG00418\_A.sort.bam \

OUTPUT=HG00418\_A.sort.rmdup.bam ASSUME\_SORTED=TRUE METRICS\_FILE=/dev/null VALIDATION\_STRINGENCY=SILENT REMOVE\_DUPLICATES=true

java -Xmx2g -jar /home/27626/bin/picard-tools-2.3.0/picard.jar MarkDuplicates INPUT=HG00418\_B.sort.bam \

OUTPUT=HG00418\_B.sort.rmdup.bam ASSUME\_SORTED=TRUE METRICS\_FILE=/dev/null VALIDATION\_STRINGENCY=SILENT REMOVE\_DUPLICATES=true

**Q1. How many reads are removed as duplicates from the files (hint view the on-screen output from the two commands)?**

It was not that many reads after all, however for eg. ancient DNA or when the library is "over-sequenced" (ie. low DNA input, many reads produced) this can easily be 20-50% or even 99% of all reads in a library.

**Merge libraries to sample level**

The rest of the analyses can run using just one bam-file pr. sample. Therefore we will merge the two libraries and index the output bam:

java -Xmx2g -jar /home/27626/bin/picard-tools-2.3.0/picard.jar MergeSamFiles \

INPUT=HG00418\_A.sort.rmdup.bam INPUT=HG00418\_B.sort.rmdup.bam OUTPUT=HG00418.sort.rmdup.bam

samtools index HG00418.sort.rmdup.bam

**Processing using GATK**

Now we start using the Genome Analysis Toolkit (GATK) developed by the Broad Institute. It is one of the leading programs for processing bam-files and SNP calling. GATK is a java program and is called as by the main program "GenomeAnalysisTK.jar" and then each function as a walker (option -T). Lets make a shortcut (symbolic link) so that the commands are easier to write. Unfortunately some of the commands are still horribly long so we will use "\" to separate the commands on several lines. Also we need to copy the chr21.fasta to here.

ln -s /home/27626/bin//GenomeAnalysisTK-3.5/GenomeAnalysisTK.jar .

cp /home/27626/exercises/alignment/human/chr21.\* .

**Realignment**

Because we are using aligners such as BWA that are optimzed to align million to billion of reads to very large genomes (the human genome is 3Gb) they are optimized for speed. Therefore some reads may be mapped wrongly, and it is known that indels (insertions and deletions) can cause wrong alignments that can lead to false SNP calls. Here we parse through the bam-file and select regions with indels in the alignment that can be realigned using a sensitive alignment algorithm.

Lets create regions that must be aligned using the RealingerTargetCreator walker. NB: All of the GATK commands will show you how far it as progressed while it is running and how much time (approx) that is left.

java -Xmx2g -jar GenomeAnalysisTK.jar \

-I HG00418.sort.rmdup.bam \

-R chr21.fasta \

-T RealignerTargetCreator \

-o IndelRealigner.intervals

Start the realignment using the IndelRealigner walker.

java -Xmx2g -jar GenomeAnalysisTK.jar \

-I HG00418.sort.rmdup.bam \

-R chr21.fasta \

-T IndelRealigner \

-targetIntervals IndelRealigner.intervals \

-o HG00418.sort.rmdup.realn.bam

**Base Quality Recalibration (Optional)**

**The recalibration of the base qualities takes some time to run and you should skip this and go to the genotyping exercise - when you have finished the genotyping exercise you can come back and do this. Another important issue is that this only works when you have human data or another species where the variation is well described.**

Now we will recalibrate the base quality scores of the reads. This is because the base qualities assigned by the machine may not truely reflect the base calling errors and will lead to potential wrong variant calls. For this we will use known SNPs from dbSNP (database of known human variants) and Mills gold standard of indels as sites that are known to be variable in humans.

We are going to recalibrate the reads based on quality scores, position in read (cycle) and dinucleotide content. First we need to count the statistics and output the sites that needs to be recalibrated. We will create files "recal\_data.grp" with the information needed to recalibrate our quality scores.

ln -s /home/27626/exercises/alignment/blast/hg38bundle/Homo\_sapiens\_assembly38.dbsnp.vcf.gz .

ln -s /home/27626/exercises/alignment/blast/hg38bundle/Homo\_sapiens\_assembly38.dbsnp.vcf.gz.tbi .

ln -s /home/27626/exercises/alignment/blast/hg38bundle/Mills\_and\_1000G\_gold\_standard.indels.hg38.vcf.gz .

ln -s /home/27626/exercises/alignment/blast/hg38bundle/Mills\_and\_1000G\_gold\_standard.indels.hg38.vcf.gz.tbi .

java -Xmx4g -jar GenomeAnalysisTK.jar \

-T BaseRecalibrator \

-R chr21.fasta \

-I HG00418.sort.rmdup.realn.bam \

-knownSites Homo\_sapiens\_assembly38.dbsnp.vcf.gz \

-knownSites Mills\_and\_1000G\_gold\_standard.indels.hg38.vcf.gz \

-nct 2 \

-o recal\_data.grp

Now we are going to recalibrate our reads in the alignment using the data that we created.

java -Xmx4g -jar GenomeAnalysisTK.jar \

-T PrintReads \

-R chr21.fasta \

-BQSR recal\_data.grp \

-nct 2 \

-I HG00418.sort.rmdup.realn.bam \

-o HG00418.sort.rmdup.realn.recal.bam

Finally lets take a look at the effect of the recalibration - we can generate a nice report using the command below (it is a little slow ~5 mins).

java -Xmx4g -jar GenomeAnalysisTK.jar \

-T BaseRecalibrator \

-R chr21.fasta \

-I HG00418.sort.rmdup.realn.recal.bam \

-knownSites Homo\_sapiens\_assembly38.dbsnp.vcf.gz \

-knownSites Mills\_and\_1000G\_gold\_standard.indels.hg38.vcf.gz \

-nct 4 \

-o recal\_data.after.grp

# plot the base recalibration

java -Xmx4g -jar GenomeAnalysisTK.jar \

-T AnalyzeCovariates \

-R chr21.fasta \

-before recal\_data.grp \

-after recal\_data.after.grp \

-plots recal.pdf

Open the pdf-file and scroll to the second page. Here you will see a lot of different plots, look at the plot in the upper left corner. This shows the reported vs. empirical quality scores at sites with mis-matches (substitutions), where pink dots are original and blue dots are the re-calibrated qualities.

**Q2. Has the recalibration worked (are the blue dots closer to the dotted line than the pink?) If not could you think of why?**

Also look at the plot in the middle row to the left. This shows the accuracy of the base qualities given the position in the read (imagine that the first and second pair going from the middle (0) to +100 and -100 respectively).

**Q3. Does this look better now after the recalibration?**

This looks ok, but we really did not have enough data (we used on chr21 and only a small part of the original data). This is what it was supposed to look like:

evince /home/27626/exercises/aln\_process/bqsr\_cycle.pdf &

evince /home/27626/exercises/aln\_process/bqsr\_quality.pdf &

evince /home/27626/exercises/aln\_process/bqsr\_dinuc.pdf &