NGS Tutorial

Roland Krause[1]

Luxembourg Centre for Systems Biomedicine (LCSB),
Unversity of Luxembourg
[1]roland.krause@uni.lu

June 11, 2014

Contents

1 Introduction

Exome sequencing is cost-effective way of sequencing by reducing the genome to the coding part by microarray hybridisation.
In this tutorial, we will map sequences using `bwa` and learn usual steps for quality improvements. In order to speed up the tutorial, only chromosome 22 is taken into account.
First you will need to extract the sequences from a given .bam file.

1.1 Set up

Create directory `ngs` for all next-generation sequencing tutorials.

```
mkdir ngs
cd ngs
```

All data source data is kept in the directory `/Users/roland.krause/Public/isb101/`.
For your convenience, create a variable holding the path to the resources.

```
RESOURCE="/Users/roland.krause/Public/isb101/"
```

Note: Not all commands are given in full in this tutorial. You might need to use the commands you have learned previously.
The programs `samtools` and `bwa` should be available from your path.
Question: How do you find out where a program is installed?

2 Extracting reads from an existing BAM file

2.1 Copy the BAM file to your local folder

This file has already been processed. We use it as source of our data. The BAM file is called `daughter.Improved.bam`
Create a *soft link* to the file.

```
ln -s $RESOURCE/daughter.Improved.bam .
```

Questions:

1. Why don't we copy the file?
2. Check the properties of the file using options of the `ls` command.
3. What happens if you would delete the link in your directory?
4. What happens if you delete the file in `$RESOURCE` ?

2.2 Index the BAM file

```
samtools index daughter.Improved.bam
```

This will take a few seconds.
Questions

1. What did the command do?
2. What is an *index*?

2.3 Visualize alignments with samtools tview

```
samtools tview \
 daughter.Improved.bam \
 human_g1k_v37_Ensembl_MT_66.fasta
```

2.4 Extract chromosome 22 from the example BAM

Slice chromosome 22 and save a piece in SAM format

```
samtools view daughter.Improved.bam 22 \
> daughter.Improved.22.sam
```

2.5 Convert SAM to FASTQ using PICARD

Create a soft link to the picard-tools directory (!) in `$RESCOURCE` in your local `ngs` directory.

```
ln -s $RESOURCE/picard-tools/ picard-tools
```

Then, run the converter as follows:

```
java -jar picard-tools/SamToFastq.jar \
 I=daughter.Improved.22.sam \
 F=daughter.22.1.fq \
 F2=daughter.22.2.fq \
 VALIDATION_STRINGENCY=SILENT
```

Inspect the output files and recapituliate the fastq-format.

3 Performing quality control of the sequenced reads

FastQC is a tool kit for quality performance. You will probably not be able to run this unless you are working from a Linux computer. If you are working from a Mac, you need to have X11 or XQuartz installed.
On the server login on a remote machine login via ssh with -X for X11 support.

```
ssh -X username@nitro.uni.lux
```

the following command opens the FastQC GUI

```
perl FastQC/fastqc
```

Load the new .sam file
We will discuss this together on screen.

4 Mapping

4.1 Indexing the reference

The following command has to be use. This step is skipped as it takes to much time. The results can be found in the `$RESOURCE` directory.

```
# bwa index -a bwtsw human_g1k_v37_Ensembl_MT_66.fasta
```

4.2 Perform alignment with Burrows-Wheeler Transform

In this main section of the mapping we will first align all reads and subsequently prepare the alignment for filtering and clean-up .
We will built indeces for further processing.
Modify the path to the reference genome in the command line.
Question: Why would a link not work in this case?

```
bwa mem -M  $RESOURCE/human_g1k_v37_Ensembl_MT_66.fasta \
daughter.22.1.fq daughter.22.2.fq \
> daughter.22.sam
```

4.3 Convert SAM to BAM

```
samtools view -bS daughter.22.sam \
> daughter.22.bam
```

4.4 Sort BAM

The suffix bam is automatically attached. This is for compatibility with PICARD and GATK.

```
samtools sort daughter.22.bam  daughter.22.sorted
```

4.5 Mark duplicate reads

Create a temporary folder and run picard tools.
Copy picard tools from the RESOURCE folder.

```
mkdir tmp

java -Djava.io.tmpdir=tmp -jar picard-tools/MarkDuplicates.jar \
 I=daughter.22.sorted.bam \
 O=daughter.22.sorted.marked.bam \
 METRICS_FILE=daughter.22.sorted.marked.metrics \
 VALIDATION_STRINGENCY=LENIENT
```

http://picard.sourceforge.net/command-line-overview.shtml
http://sourceforge.net/apps/mediawiki/picard/index.php?title=Main_Page

4.6 Add read-group

This step is only necesssary for data generated around 2012.

```
java -jar picard-tools/AddOrReplaceReadGroups.jar \
INPUT=daughter.22.sorted.marked.bam \
OUTPUT=daughter.22.prepared.bam \
RGID=group1 RGLB= lib1 RGPL=illumina RGPU=unit1 RGSM=sample1

java -jar picard-tools/BuildBamIndex.jar INPUT=daughter.22.prepared.bam
```

5 Quality improvements

Index the file with picard (Step A).

```
java -Xmx4g -jar picard-tools/CreateSequenceDictionary.jar \
R=human_g1k_v37_Ensembl_MT_66.fasta \
O=human_g1k_v37_Ensembl_MT_66.dict
```

Create an index on the reference sequence using samtools.

```
samtools faidx human_g1k_v37_Ensembl_MT_66.fasta
```

5.1 Realignment

Create a index using GATK.
Note that we use java7 for running GATK, required for the latest version.

```
java7 -Xmx4g -jar GenomeAnalysisTK.jar \
-T RealignerTargetCreator \
-R human_g1k_v37_Ensembl_MT_66.fasta \
-o daughter.bam.list \
-I daughter.22.prepared.bam

java7 -Xmx4g -Djava.io.tmpdir=./tmp/ -jar GenomeAnalysisTK.jar \
-T IndelRealigner \
-I daughter.22.prepared.bam \
-R human_g1k_v37_Ensembl_MT_66.fasta \
-targetIntervals daughter.bam.list -o daughter.22.real.bam
```

5.2 BQSR(Base Quality Score Recalibration)

```
java7 -Xmx4g -jar GenomeAnalysisTK.jar \
  -T BaseRecalibrator \
  -I daughter.22.real.bam \
  -R human_g1k_v37_Ensembl_MT_66.fasta \
  -knownSites dbsnp_135.b37.vcf \
  -o recal_data.table
```

5.3 Fix the mate pairs

```
java7 -Djava.io.tmpdir=./tmp -jar picard-tools//FixMateInformation.jar INPUT=daughter.22.rea
```

6 Variant calling

The final step of variant calling with the two tools most often used, `samtools` and `GATK`.

6.1 Samtools mpileup

```
samtools mpileup \
 -S -E -g -Q 13 -q 20 \
 -f human_g1k_v37_Ensembl_MT_66.fasta \
 daughter.22.real.bam | \
 bcftools \
 view -vc - > daughter.22.mpileup.vcf
```

Note: not working at the moment.

6.2 GATK Unified Genotyper

```
java7 -Djava.io.tmpdir=tmp -jar GenomeAnalysisTK.jar \
 -l INFO \
```

```
-T UnifiedGenotyper \
-R human_g1k_v37_Ensembl_MT_66.fasta \
-I daughter.22.real.bam \
-stand_call_conf 30.0 \
-stand_emit_conf 10.0 \
--genotype_likelihoods_model BOTH \
--min_base_quality_score 13 \
--max_alternate_alleles 3  \
-A MappingQualityRankSumTest \
-A AlleleBalance \
-A BaseCounts \
-A ChromosomeCounts \
-A QualByDepth \
-A ReadPosRankSumTest \
-A MappingQualityZeroBySample \
-A HaplotypeScore \
-A LowMQ \
-A RMSMappingQuality \
-A BaseQualityRankSumTest \
-L 22 \
-o daughter.22.gatk.vcf
```

Questions:

1. How many variants are called?
2. Are both callers come up with the same variants?
3. Inspect a case for indels and SNP and check those variants using `samtools tview`. Take screen shots.

Next steps: Annotation and comparison of samples.
Acknowledgement: Holger Thiele, Kamel Jabbari (CCG Cologne), Patrick May, Dheeraj Bobbili (LCSB)