Exercises NGS Tutorial – Part 2

Stefan Wyder
URPP Evolution
University of Zurich

Feb 2014

Today's data files are available in the directory /home/studi15/EcoliDB10 on the server 130.60.201.40 (only visible from the UZH network – log in via VPN if necessary). It is more convenient to work locally on your laptop. In case this is not possible you can work on the server.

An alternative to download the files:
https://dl.dropboxusercontent.com/u/3435091/EcoliDH10B.fa
https://dl.dropboxusercontent.com/u/3435091/EcoliDH10B.gff
https://dl.dropboxusercontent.com/u/3435091/install_software_tutorial_NGS_Part2.sh
https://dl.dropboxusercontent.com/u/3435091/MiSeq_Ecoli_DH10B_110721_PF_subsample.bam

To work locally on your laptop
The following instructions have been tested on Ubuntu. Mac users can use Ubuntu in the Virtual Machine setup I distributed for the last session.

First copy the files to your computer

```
scp -r studi15@130.60.201.40:~/EcoliDB10/ .
```

Then install the required software (freebayes, bedtools, vcftools and their dependencies). If you are using Ubuntu simply run the shell script install_software_tutorial_NGS_Part2.sh
It contains the following lines (check before you run it!):

```
sudo apt-get install vcftools bedtools samtools git cmake g++ zlib1g-dev
cd
cd software
git clone --recursive git://github.com/ekg/freebayes.git
cd freebayes
make
```

Now you can start with exercise 1.

To work on the server. Log in to your account typing `ssh studiX@130.60.201.40` where X is the number 1-14 assigned to you. Please do not work as studi15. The data files are then on /home/studi15/EcoliDB10 and the programs we will use today are installed under /home/studi15/software/

Copy the files to your computer

```
scp -r studi15@130.60.201.40:~/EcoliDB10/ .
```

then start with exercise 1.

Exercise 1: Experiment Setup and BAM file

Today we are working with genome resequencing data for *E. coli* DB10 strain (2x150 bp paired end from Illumina MiSeq). The reads were aligned to the *E. coli* DB10 reference genome (4.7 MB sequence length, http://www.ncbi.nlm.nih.gov/nuccore/NC_010473.1). We want to use it to perform variant calling.

To save disk space a random subset of the BAM (Binary Alignment Map) file has already been prepared for you. It is about 1/40 of the complete BAM file. Now we have an average genome coverage of approx. 40x, i.e. on average for every nucleotide of the genome you will find 40 aligned reads. Unmapped reads were removed from the BAM to decrease the size and it was position-sorted (samtools sort).

A BAM file includes mapping information as well as nucleotide sequence and quality for each read. Try to understand the format - we discussed it in the last session.
You can also consult the specification for details: http://samtools.sourceforge.net/SAMv1.pdf

> Explore the BAM file with “samtools view BAMFILE | less”. Also use the option “-S” for less that prevents line wraps, so you can see one alignment per line.

Exercise 2: Quality control

Like in the last session we are using the program FastQC for the quality control. By default it provides its quality report in the form of a webpage, which you can open in your browser.
For launching fastqc type

```
$ ~/software/FastQC/fastqc BAMFILE
```

The output files are by default created in same directory than BAMFILE in a directory called BAMFILE_fastqc. It contains a file fastqc_report.html that you can your web browser. (Under Linux simply type: firefox fastqc_report.html on the command line).

Go through the different graphs and try to understand them.

Exercise 3: BAM file preprocessing

BAM File preprocessing is key to do reliable SNP calling. Possible steps include (For details check the GATK documentation):

1. Filtering anomalous read pairs (e.g. with unexpected insert size, read pairs where only 1 read could be mapped, …) using samtools and other tools)
2. Mark Duplicate Removal (e.g. samtools or Picard)
3. Indel Realignment (e.g. using Picard)
4. Base Recalibration (e.g. using Picard)

Depending on the SNP Caller used we have to perform all or just a subset of preprocessing steps. Luckily, FreeBayes does the indel realignment step internally and does not require base recalibration. We just have to do steps 1 + 2.
In our sample we observe very few anomalous read pairs, so we will skip step 1. Lets just mark duplicate reads, Reads marked as duplicates in the BAM file will then be ignored by FreeBayes:

```
$ samtools rmdup BAMFILE BAMFILE_dedup.bam
```

Now we have an analysis-ready BAM.

Then we create an index file for the bam file:

```
$ samtools index BAMFILE_dedup.bam
```

The index command creates an additional file with which genomic coordinates can quickly be translated into file offsets for faster access. We will need BAM index files for the next exercise where we visualize reads.

Exercise 4: Visualize the aligned reads

Seeing is believing! One should always have at look at the data to get a feeling about the error rate, coverage heterogeneity, …

Go to the Integrative Genome Viewer (IGV) website http://www.broadinstitute.org/igv/

1. Go to the download site and register

2. Launch IGV
3. Load the fasta file of the genome: File | Load Genome from File…
   then choose the file EcoliDH10B.fa
4. Load the BAM file: File | Load from File…
   then choose the BAM file
5. Load the genome annotation (gff or bed): File | Load from File…
   then choose the file EcoliDH10B.fa

Try to understand the different windows, colors etc you see. Use the help to get information. Go through the menus to get an idea about the functionality.

Move across some regions. What type of differences from the reference do you see?

Inspect the coverage heterogeneity.

Exercise 5: SNP Callling

Here we use FreeBayes, which is a fast easy-to-run variant caller for short variants (Garrison & Marth, http://arxiv.org/abs/1207.3907). It identifies small polymorphisms, specifically SNPs (single-nucleotide polymorphisms), small indels (insertions and deletions), MNPs (multi-nucleotide polymorphisms), and complex events (composite insertion and substitution events) smaller than the length of a short-read sequencing alignment. GATK is another widely used short variant caller that was developed for the human "1000 genomes" project. GATK is slow/CPU-intensive to run and a relatively complex pipeline but it is becoming de facto standard at least for model organisms. Both short variant callers identify only short variants (in practice up to maybe 30 bp) relative to a reference sequence. If we want to identify larger polymorphisms we have to use different specialized software.

Here we will identify short variants relative to the genome of the *E.coli* DB10 strain (Ncbi NC_010473). The experiment was done with the same strain the reference genome was made so we expect only a very small number of genetic differences. It is a relatively easy task – *E. coli* is haploid, the genome does not contain long repeats and we have a homogenous coverage across almost the whole genome. In many cases it will be much harder to identify SNPs reliably e.g. for diploid eukaryotic genomes containing lots of repetitive elements, pseudogenes and close paralogs.

FreeBayes expects a cleaned BAM file as input with marked duplicates. It calls variants for any number of individuals of a population, we can combine data from multiple individuals into a single BAM by attaching Read Group (RG) to the alignments.

Let us run FreeBayes:

```
PathFreeBayes="~/software/freebayes/bin"
$ $PathFreeBayes/freebayes --help
```

We can run an analysis without any parameter optimization just setting the ploidy to 1:

```
$ $PathFreeBayes/freebayes -p 1 -f EcoliDH10B.fa BAMFILE_dedup.bam >
BAMFILE.vcf
```

Inspect the vcf output file and try to understand what the different columns mean. You will find a very nice poster summary under http://vcftools.sourceforge.net/VCF-poster.pdf . The detailed vcf format specifications you find under http://www.1000genomes.org/wiki/Analysis/Variant%20Call%20Format/vcf-variant-call-format-version-41

Different metrics can be used to filter the variants, the FreeBayes manual states: Of primary interest to most users is the QUAL field, which estimates the probability that there is a polymorphism at the loci described by the record. In freebayes, this value can be understood as 1 - P(locus is homozygous given the data). It is recommended that users use this value to filter their results, rather than accepting anything output by freebayes as ground truth. By default, records are output even if they have very low probability of variation, in expectation that the VCF will be filtered using tools such as vcftools.

Check what are the options of FreeBayes. Use the FreeBayes manual (https://github.com/ekg/freebayes) to explore and change parameters. Important parameters are --min-mapping-quality and --min-base-quality. How do different parameters influence your results?

How would you filter? Which cut-off would you use?
Sort the records according to quality and check some using the IGV genome browser.
Hint: you can sort the vcf file using

```
grep -v "#" BAMFile.vcf | sort -k6,6gr | head
```

Do you think some records are SNPs? Or simply sequencing errors? Look up the base call qualities at the varying positions by moving the cursor to a nucleotide.

Exercise 6: VCFtools

VCFtools are specialized tools for working with VCF files: validating, filtering merging, comparing and calculate some basic population genetic statistics (see the documentation http://vcftools.sourceforge.net/docs.html). There are many other tools with similar functionality available for the same purpose.

Exercise 7: How many low-coverage regions?

Often some regions of the genome are low coverage only (or even without any aligned reads) consequently we cannot tell whether polymorphisms exist in these regions. We want to identify such regions and find out whether they overlap with genes.

The **bedtools** utilities are convenient for working with genomic coordinates for example bedtools allows one to intersect, merge, count, complement, and shuffle genomic intervals from multiple files in widely-used genomic file formats such as BAM, BED, GFF/GTF, VCF. You will find the documentation under http://bedtools.readthedocs.org/en/latest/index.html

Try to find out how many nucleotides in the *E. coli* genome do not reach a minimal read coverage of 5, 10 or 20 reads.
(Hint: Use samtools depth, direct the output into a file and then use awk or R to process the file)

Do some low coverage nucleotides overlap with annotated genes?
(Hint: use the command intersect from bedtools)

We now counted individual nucleotides but actually we would like to know whether there are some regions with low coverage
(Hint: use mergeBed from bedtools)

Are there any regions with unexpected high coverage? As mapping problems are likely (due to repeats) they are often excluded from the analysis.