Exercises NGS Tutorial – Part 1

Stefan Wyder
URPP Evolution
University of Zurich

Dec 2013

Today's data files are available in the directory /home/studi15/Morax on the server 130.60.201.40. You can either work on the server using your account or locally on your laptop.

To work locally copy the files to your computer

```
scp studi15@130.60.201.40:~/Morax/* .
```

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/Morax and the programs we use today are installed under /home/studi15/software/

Exercise 1: FASTQ data format

The standard format from most next-generation-sequencers is FASTQ. Here we are working with data from an Illumina HiSeq 2000. Currently 1 lane will usually produce approx. 190 mio reads (88 Gb of sequence for 2x100 bp) and thus a full run (2 flowcells with 8 lanes each) will usually produce 3 bio reads (approx. 600 Gb of sequence for 2x100bp) with a running time of about 2 weeks. FASTQ files are compressed (gzipped) to save disk space. First look at the RNA-seq data sample stored in the Exp1_26C_CGATGT_L008_R1_001.fastq.gz using zmore or zless. It is about 1/5th of a lane.

1. This file includes qualities for each nucleotide. Try to understand the format
   You can also consult wikipedia for details: http://en.wikipedia.org/wiki/FASTQ_format
2. How many reads does it contain?

Exercise 2: Quality control

If we get new data we first want to assess the quality of the data. We can use the program FastQC for the quality control. Here we will use it in the interactive mode with a graphical user interface (but it can also be used on the command line – handy for many files). By default it will provide its quality report in the form of a webpage, which you can open in your browser.

First install FastQC from its webpage
http://www.bioinformatics.babraham.ac.uk/projects/fastqc/
Choose the file to download according to your operating system. If you get stuck with installation check the 'Installation and setup instructions' on the download page.

1. Run FastQC and choose the file containing the FASTQ reads. Go through the different graphs and try to understand them.
2. Try to find out the maximum read length.
3. Look up the median base call quality for the reads at positions 1, 50 and 100 and calculate how many errors you will expect at the respective positions.

Exercise 2: Download and install bowtie2 software

Bowtie2 is a short-sequence read aligner (e.g. 150nt long). The reads are aligned to a reference sequence (e.g. human genome).

Install bowtie2.

Mac OS X users only:
if you have installed a package manager, use it to install bowtie2, e.g. for homebrew:

```
brew search bowtie2
sudo install homebrew/science/bowtie2
```

If you don’t have available a package manager, go to the respective webpage http://bowtie-bio.sourceforge.net/bowtie2/index.shtml and download the binary for Mac OS X (called bowtie2-2.1.0-macos-x86_64.zip).

Linux users only:
Possibly you have compiled bowtie2 in Part 2 of the Linux tutorial. Then you can go on to use it.

Try to use the package manager to install bowtie2, e.g. for Ubuntu:

```
apt-cache search bowtie2
sudo apt-get install PackageName
```

where *PackageName* is the exact name of the package

Alternatively go to the respective webpage
http://bowtie-bio.sourceforge.net/bowtie2/index.shtml and download the binary for Linux.

You can now try to run bowtie2 being in the directory the binary bowtie2 is located in:

```
$ ./bowtie2
```

Optional: Being able to type "bowtie2" in any directory to run in:
Since bowtie2 directory is not in the $PATH environment variable (a list of directory locations which Unix searches for commands when you try to run them), you can only run it from the bowtie2-2.1.0 folder or by providing the full path (somethink like e.g under Linux: `/home/swyder/software/bowtie/bowtie2-2.1.0/bowtie2` or under Mac OS X: `/Users/swyder/software/bowtie2-2.1.0/bowtie2`). You can add the bowtie2 folder to the $PATH:

```
$ export PATH=$PATH:/home/username/software/bowtie2-2.1.0
```

Now you can simply type "`bowtie2`" anywhere (in any directory) and the **shell** will find the **bowtie2** software. The modification to $PATH affects only the current window until it is closed (to make it permanent you would have to add it to ~/.bash_profile).

Exercise 3: Align reads to Moraxella genome

Moraxella catarrhalis (http://www.ncbi.nlm.nih.gov/nuccore/NC_014147) is an interesting Gram-negative Gammaproteobacterium and a human pathogen of the respiratory tract. The whole genome sequence is already available on the server in the `Morax/Moraxella_catarrhalis_O35E.fasta` file. The FASTA format is widely used in sequence distribution, see the description at http://en.wikipedia.org/wiki/FASTA_format.

Now we are ready to use **bowtie2**. First build the index of the *Moraxella* reference genome. The format is: "`bowtie2-build <fasta_file> <custom_index_name>`". In the Morax folder, you could use:

```
$ bowtie2-build Moraxella_catarrhalis_O35E.fasta Morax
```

Use **grep** to find out how many contigs are present in the file. How large is the genome?

Once the index is created (you do this only once for each reference genome, i.e. each FASTA file), you align one read (sequence) to the genome by typing:

```
$ bowtie2 Morax -ac CTGTATCACCGATTT > Morax.sam
```

Explore the SAM results file with “`less -S`”. The parameter “-S” prevents line wraps, so you can see one alignment per line.

Exercise 4: Alignment of RNA-seq sample reads to the *Moraxella catarrhalis* genome

To align the RNA-seq reads in the Exp1_26C_CGATGT_L008_R1_001.fastq.gz file, you first need to index the *Moraxella catarrhalis* genome.

After you created the index (Exercise 3), you can align the reads by typing:

```
$ bowtie2 Morax -U Exp1_26C_CGATGT_L008_R1_001.fastq.gz > Morax2.sam
```

The results are returned in SAM format and stored to the Morax2.sam file. How many reads align?

Check what are the options of bowtie2. Use Bowtie2 manual (http://bowtie-bio.sourceforge.net/bowtie2/manual.shtml) to explore and change parameters. How do different parameters influence your results?

Exercise 5: Displaying the alignment using samtools

First install samtools.

Mac OS X users only:
Unfortunately the samtools developers don’t provide binaries for Mac OS X.
If you have installed a package manager, use it to install samtools, e.g. for homebrew:
sudo install samtools

If you don’t have available a package manager but you have installed a C compiler (Part 2 of the Linux tutorial), go to the respective webpage http://samtools.sourceforge.net/ , download the source code and compile it.

Otherwise you can give it try with the binary I compiled for OS X 10.8.5 (no guarantee it will work!!)

```
$ scp studi15@130.60.201.40:~/samtools* .
```

Linux users only:
Like always first try whether its available through the package manager
apt-cache search samtools
sudo apt-get install *PackageName*

In case its not available go to the respective webpage http://samtools.sourceforge.net/ , download the source code and compile it.

The **samtools** are the main tool to work with SAM and BAM files. The samtools allow to convert SAM and BAM files as well as viewing, indexing, filtering, sorting, and merging BAM files. These commands are often used and should be installed on every system. Documentation is relatively sparse, see http://samtools.sourceforge.net/samtools.shtml . Simply typing

```
$ samtools
```

displays the available commands. The main commands are view, sort, merge and rmdub. By typing samtools <command>, e.g.

```
$ samtools view
```

you get the available options for viewing BAM files.

Lets start the work. First we want to do the conversion sam -> bam file and sort the bam file. (samtools sort will append the extension .bam to the output name Morax2_sorted)

```
$ samtools view -bS Morax2.sam | samtools sort - Morax2_sorted
```

The above command is just piping together the following 2 lines without writing the temporary file Morax2.bam to disk

```
$ samtools view -bS Morax2.sam > Morax2.bam
$ samtools sort Morax2.bam Morax2_sorted
```

BAM files save disk space and allow fast access to selected regions and reads. Importantly, SAM and BAM files contain all input reads including the ones which could not be mapped to the genome! The <samtools sort> command sorts the reads according to coordinates for faster access. Many downstream applications require sorted bam files.

Now we can delete the sam file as all the information is contained in the bam file.

```
$ rm -i Morax2.sam
```

Then we create an index file for the bam file:

```
$ samtools index Morax2_sorted.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.

In addition to the above mentioned commands, the samtools also comprise commands to call SNPs and indels from BAM files which we will use at a later stage.

📝Exercise 6: 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/

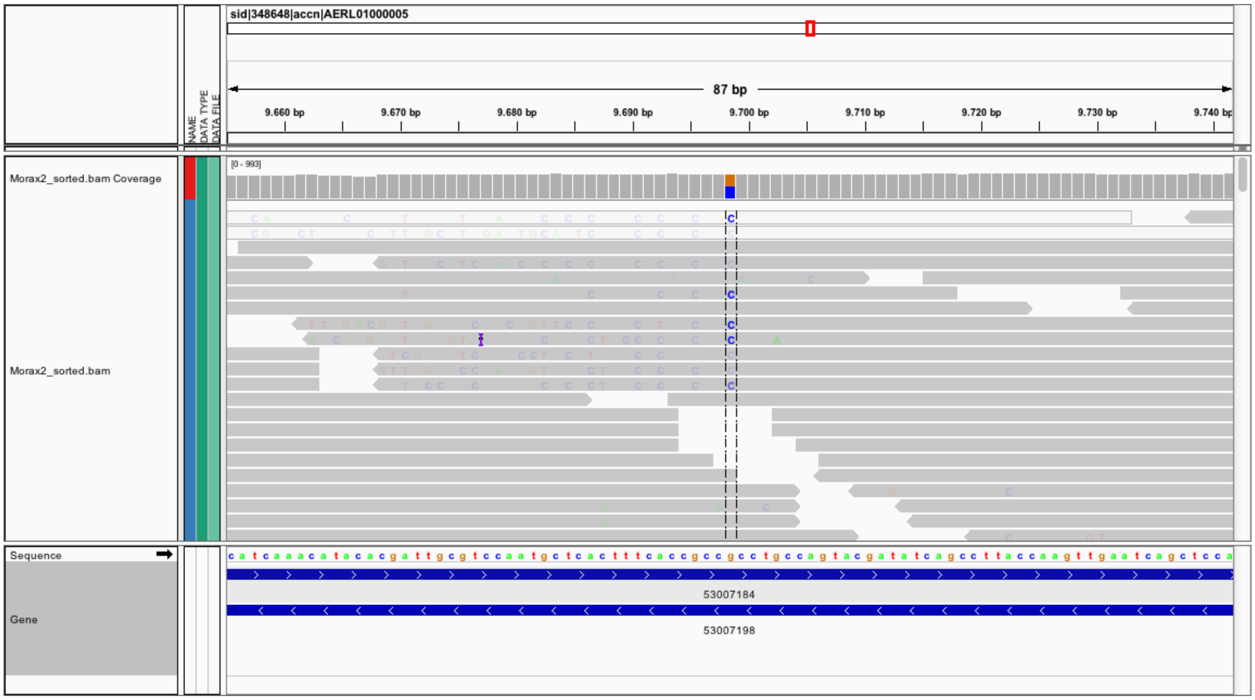
1. Go to the download site and register
2. Launch IGV
3. Load the fasta file of the genome: File | Load Genome from File…
   then choose the file Moraxella_catarrhalis_O35E.fasta
4. Load the BAM file: File | Load from File…
   then choose the BAM file
5. Load the genome annotation (gff or bed): File | Load from File…
   then choose the file Moraxella_catarrhalis_O35E.PATRIC.gff

❓ 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.

❓ Go to the following positions (copy to the IGV position window):
sid|338577|accn|AERL01000003:58,447-58,673
sid|348648|accn|AERL01000005:9,655-9,740
sid|338577|accn|AERL01000003:94,397-94,509
Do you think they contain SNPs? Or are there simply sequencing errors? Look up the base call qualities at the varying positons by moving the cursor to a nucleotide.


sid|348648|accn|AERL01000005
87 bp
9.660 bp
9.670 bp
9.680 bp
9.690 bp
9.700 bp
9.710 bp
9.720 bp
9.730 bp
9.740 bp
NAME
DATA TYPE
DATA FILE
Morax2_sorted.bam Coverage
[0 - 993]
Morax2_sorted.bam
Sequence
c a t c a a a c a t a c a c g a t t g c g t c c a a t g c t c a c t t t c a c c g c c g c c t g c c a g t a c g a t a t c a g c c t t a c c a a g t t g a a t c a g c t c c a
Gene
53007184
53007198