ChIP-Seq data analysis workshop

Exercise 1. ChIP-Seq peak calling

1. Using Putty (Windows) or Terminal (Mac) to connect to your assigned computer.

Create a directory /workdir/myUserID (replace myUserID with you BioHPC ID), copy the fastq, reference genome sequence (Arabidopsis_thaliana.TAIR10.31.dna.genome.fa), reference genome index files and bam files to the working directory.

```
mkdir  /workdir/myUserID
cd  /workdir/myUserID
cp  /shared_data/ChIP_seq_workshop_2017/*.fastq  ./
cp  /shared_data/ChIP_seq_workshop_2017/ Arabidopsis_thaliana* ./
cp  /shared_data/ChIP_seq_workshop_2017/ *.bam* ./
cp  /shared_data/ChIP_seq_workshop_2017/ *.pl ./
cp  /shared_data/ChIP_seq_workshop_2017/ test* ./
```

2. Check sequencing quality from bam or fastq file, in this test sample, bam file used as input. There are two ways in which FASTQC can be run in "command line" mode, or as a GUI (graphical user interface). The GUI needs Xming locally installed and opened, otherwise firefox command doesn't work. In this example, you can run fastqc in command mode and copy results (a html report with a nice graphical summary output and a compressed folder) to your local terminal by filezilla.

```
fastqc treatment.fastq
```

3. FASTX-Toolkit (http://hannonlab.cshl.edu/fastx_toolkit/) to filter and trim sequences based on quality. The FASTX-Toolkit is a collection of command line tools for preprocessing of raw fastq files.

There are many tools available within FASTX-toolkit, we will be using two of those tools:

• fastq_quality_filter: Filters sequences based on quality

• fastx_trimmer: Shortening reads in a FASTQ or FASTQ files (removing barcodes or noise).

For most functions of programs, you can see their instructions by typing their name in the terminal followed by the flag -h. There are many options available, and we will use two in this practice.

fastq_quality_filter -v -Q 33 -q 20 -p 80 -i treatment.fastq -o treatment_quality_filter.fastq

```
  [-h]         = This helpful help screen.
  [-q N]       = Minimum quality score to keep.
  [-p N]       = Minimum percent of bases that must have [-q] quality.
  [-z]         = Compress output with GZIP.
  [-i INFILE]  = FASTA/Q input file. default is STDIN.
  [-o OUTFILE] = FASTA/Q output file. default is STDOUT.
  [-v]         = Verbose - report number of sequences.
```

```
fastx_trimmer -Q 33 -f 3 -l 40 -m 30 -i treatment.fastq -o
treatment_trim.fastq

  [-h]         = This helpful help screen.

  [-f N]       = First base to keep. Default is 1 (=first base).

  [-l N]       = Last base to keep. Default is entire read.

  [-t N]       = Trim N nucleotides from the end of the read.

  [-m MINLEN]  = With [-t], discard reads shorter than MINLEN.
```

4. ChIPQC package was used for the rapid generation of ChIP-seq quality metrics from aligned data in BAM format. It includes metrics as following:

Reads total number of reads in the bam file.

Map% Percentage of total reads that were successfully mapped (aligned).

Filt% Percentage of mapped reads passing MapQ filter, in this case having a mapping quality score less than or equal to 15 (mapQCth=15 by default)

Dup% Percentage of mapped reads marked as duplicates.

ReadL Mean read length (as integer) derived from the data.

FragL Predicted fragment length by cross-coverage method. The fragment length is estimated by methods implemented in the chipseq package by systematically shifting the reads on each strand towards each other.

RelCC The relative cross-coverage score. The RelativeCC metric is calculated by comparing the maximum cross coverage peak (at the shift size corresponding to the fragment length) to the cross coverage at a shift size

corresponding to the read length, with higher scores (generally 1 or greater) indicating good enrichment.

The cross-coverage scores after successive shifts can then be visualized to identify the expected increase in cross-coverage scores around the fragment length as well as any evidence of artefacts by a peak in the cross-coverage score at the read length. Copy results to local computer to check it.

```
R
source("https://bioconductor.org/biocLite.R")
biocLite(c("AnnotationDbi"))
library("AnnotationDbi")
biocLite(c("AnnotationForge"))
library("AnnotationForge")
biocLite(c("GenomicAlignments", "ChIPQC"))
library(GenomicAlignments)
library(ChIPQC)
bamFile <- file.path(getwd(),
"./treatment2.fastq_filter_qual_bowtie2_sorted_filter_sorted.bam")
treatment_out = ChIPQCsample(bamFile,peaks=NULL,annotation=NULL)
QCmetrics(treatment_out)
pdf(file="cross-correlation.pdf")
plotCC(treatment_out)
dev.off()
```

5. Calling peaks with MACS2

MACS takes mapped BAM files of ChIP-seq and control samples and calls peaks. To call peaks, we will use the main module in MACS2 called 'callpeak'

```
source /programs/bin/util/setup_macs2.sh
macs2 predictd –i
treatment2.fastq_filter_qual_bowtie2_sorted_filter_sorted.bam  -g 1.0e+8  -m 10 30
macs2 callpeak  -t
treatment.fastq_filter_qual_bowtie2_sorted_filter_sorted.bam -c
control.fastq_filter_qual_bowtie2_sorted_filter_sorted.bam -g 1.0e+8  -n
test_results --nomodel --shift 0 --extsize 300 -m 10 30
```