BCGES short courses, session 1
Introduction

The point of this first session is to introduce the tools that we will use to manipulate sequence data. Generally speaking, we will deal with either `R`, bash scripting or Galaxy.

`R` is not usually the preferred way to deal with NGS data. One reason for this is that the general approach of `R` is to load data into the RAM as a first instance, which is not practical when the datasets are very large. Shell scripts (aka command line tools) that read the data on a line per line basis are usually preferred. Nevertheless, `R` has developed tools to overcome these limitations and there is, in fact, surprisingly much that one can do with it. It is also a practical tools for teaching purposes, which is useful in the context of these short courses. So we will use `R` mostly to play with the data and show the sort of things one may want to do with it.

Contents

1 Basic fastq reading and quality control

1.1 Basic shell scripting to read NGS files

1.2 Using the shortRead package in R

We start by loading one of the most relevant library, called "ShortReads". This package may not be installed but it can easily done so by running:

```
source("http://bioconductor.org/biocLite.R")
biocLite("ShortRead")
```

```
library('ShortRead')
```

As a starting point it is possible to read some of the examples fastq files and create relevant R objects.

```
fastq1.1 <- readFastq('../../data/fastq_files/fastq1_1.txt')
fastq1.2 <- readFastq('../../data/fastq_files/fastq1_2.txt')
```

We can now display the sequences and the qualities. Note that specific classes have been defined to store each of these objects. Much work has gone into figuring out how to do this.

```
reads <- sread(fastq1.1)
class(reads)

## [1] "DNAStringSet"
## attr(,"package")
## [1] "Biostrings"

head(as.character(reads))

## [1] "NACCACTCAGCTCTGGCCAATTATTGCCGTGCAGGAGTGTGGGCTCCTAGTGGCAGGGGGTCTGGAACTGTGGAAGAAGCAGGCAAACGC"
## [2] "NTCCCCAAAGCACAGGGCTCAGCTCCAGAGGGAGACGGGCTGGGCTGTCAGCGGGCCCAGGGGCACGCCACTGTTCAGAACAACTGGTTG"
## [3] "NCTATGGACTGTGGTAAAGCTAGGATTAGTAACCAGACATTACTTACCTTGGCTCCGATCTGGTTGCCACACTGGCCAATCTGAATATGG"
## [4] "NACTGAAAAAGGATGCTTTGGAAAAAGAAAGTGGGTCTGGCAACACTGACTCAACCTTGAATTCCCCGCACGATGACACGGATGACAGGG"
## [5] "GTTATTTAAGCCACCCAGTCTGTGTTTGTTATGGCAGGCTGAGGAGACTATGACAGGAACCAAACACAAAAAAAACCAAAACTCTGGAGG"
## [6] "ATTTTGTAAACCTCTTATCCTTAGATCCAAAAGATATGTTCATCTAGGCTTGATAAGCACATGTGCATTTATACCACACTCTATAGTTCT"

ids <- id(fastq1.1)
class(ids)

## [1] "BStringSet"
## attr(,"package")
## [1] "Biostrings"

head(as.character(ids))

## [1] "A81CH8ABXX:4:1101:1524:1813#NGACCAAT/1"
## [2] "A81CH8ABXX:4:1101:1583:1836#NGACCAAT/1"
## [3] "A81CH8ABXX:4:1101:1729:1852#TGACCAAT/1"
## [4] "A81CH8ABXX:4:1101:1642:1867#TGACCAAT/1"
## [5] "A81CH8ABXX:4:1101:1715:1891#TGACCAAT/1"
## [6] "A81CH8ABXX:4:1101:1624:1941#TGACCAAT/1"
```

2 Reading and interpreting quality scores

Of particular interest to us are the quality scores and we can keep using R to see what they look like:

```
quals <- quality(fastq1.1)
class(quals)

## [1] "SFastqQuality"
## attr(,"package")
## [1] "ShortRead"

quals

## class: SFastqQuality
## quality:
##   A BStringSet instance of length 2500
##         width seq
##    [1]    90 BQXXQY[V[Yccc_c__V\_ccc___\\[X...BBBBBBBBBBBBBBBBBBBBBBBBBBBBB
##    [2]    90 BBBBBBBBBBBBBBBBBBBBBBBBBBBBBB...BBBBBBBBBBBBBBBBBBBBBBBBBBBBB
##    [3]    90 BWSSQVUVUTTQVSVUUUWW___BBBBBBB...BBBBBBBBBBBBBBBBBBBBBBBBBBBBB
##    [4]    90 BBBBBBBBBBBBBBBBBBBBBBBBBBBBBB...BBBBBBBBBBBBBBBBBBBBBBBBBBBBB
##    [5]    90 gefgggggdgegfgggdfegee`eedddbd...BBBBBBBBBBBBBBBBBBBBBBBBBBBBB
##    ...   ... ...
## [2496]    90 gfgggggggegggggggfggefefggfcga...edeed0Z^``IZXYW^\YW]`ccX_adbdb
## [2497]    90 ggggggggggggggfgggggggdgfgcgg...ecgggegdfeeeg^dbdee`bNddeee^a
## [2498]    90 gggggggggggggggggggggggggggggg...e[^d[_Y``b]a[bZXZ[Y`U^BBBBBBB
## [2499]    90 eedeee]decdbdbacca`bdbbdac_adW...Wbbd__aa[dee`e\bb_cad`dZcZc\b
## [2500]    90 gggggfggggggggggdfgdfggggggg...fgegggcgg^gggeffedfgggfeggeeg
```

The ShortRead package will attempt to guess what these quality scores mean, for example see:

```
encoding(quality(fastq1.1))

##  ;  <  =  >  ?  @  A  B  C  D  E  F  G  H  I  J  K  L  M  N  O  P  Q  R  S
## -5 -4 -3 -2 -1  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
##  T  U  V  W  X  Y  Z  [ \\  ]  ^  _  `  a  b  c  d  e  f  g  h  i
## 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41

fastq2.1 <- readFastq('../../data/fastq_files/fastq2_1.txt')
encoding(quality(fastq2.1))

##  !  "  #  $  %  &  '  (  )  *  +  ,  -  .  /  0  1  2  3  4  5  6  7  8  9
##  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
##  :  ;  <  =  >  ?  @  A  B  C  D  E  F  G  H  I  J
## 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41
```

2.1 Using the Galaxy server

Start by identifying a working instance of the Galaxy server from this location. There are multiple choices here, so feel free to experiment. In case of doubt, I used the following server: http://biominavm-galaxy.biomina.be/galaxy/.
Exercise: Create an account on one of these Galaxy servers and upload the pair of fastq *fastq2_1.txt* and *fastq2_2.txt*.

2.2 Basic shell scripting to parse fastq files

3 Merging overlapping with paired reads

4 Session info

```
sessionInfo()

## R version 3.1.0 (2014-04-10)
## Platform: x86_64-unknown-linux-gnu (64-bit)
##
## locale:
##  [1] LC_CTYPE=en_US.iso885915       LC_NUMERIC=C
##  [3] LC_TIME=en_US.iso885915        LC_COLLATE=en_US.iso885915
##  [5] LC_MONETARY=en_US.iso885915    LC_MESSAGES=en_US.iso885915
##  [7] LC_PAPER=en_US.iso885915       LC_NAME=C
##  [9] LC_ADDRESS=C                   LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.iso885915 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] parallel  stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] knitr_1.6            ShortRead_1.20.0     Rsamtools_1.14.3
## [4] lattice_0.20-29      Biostrings_2.30.1    GenomicRanges_1.14.4
## [7] XVector_0.2.0        IRanges_1.20.7       BiocGenerics_0.8.0
##
## loaded via a namespace (and not attached):
##  [1] Biobase_2.22.0      bitops_1.0-6        evaluate_0.5.5
##  [4] formatR_0.10        grid_3.1.0          highr_0.3
##  [7] hwriter_1.3         latticeExtra_0.6-26 RColorBrewer_1.0-5
## [10] stats4_3.1.0        stringr_0.6.2       tools_3.1.0
## [13] zlibbioc_1.8.0
```