Evolution and genomics, Cesky Krumlov

Daniel Berner

2. February 2016

# Genomic analyses using RADseq:

# 1. Raw data manipulation

# Demo

Upload and inspection of a raw Illumina sequence data set (stickleback RAD sequences from the Misty system, Canada, unpublished; 100k subset from a single SE100 Illumina lane)

```
library(ShortRead)
# d<-readFastq(dirPath='C:/Users/daniel/Documents/science/teaching/cesky
# krumlov 2016/course.materials/R.files',
# pattern='illumina.SE100.fastq', withIds=T)
d


## class: ShortReadQ
## length: 100000 reads; width: 100 cycles


# Call the read IDs of the ShortRead object
id(d)[1:4]  # just the first four elements


##   A BStringSet instance of length 4
##     width seq
## [1]    53 BS-DSFCONTROL03:312:C3...1101:1232:2073 1:Y:0:
## [2]    53 BS-DSFCONTROL03:312:C3...1101:1213:2079 1:Y:0:
## [3]    53 BS-DSFCONTROL03:312:C3...1101:1185:2091 1:Y:0:
## [4]    53 BS-DSFCONTROL03:312:C3...1101:1102:2093 1:Y:0:
```

```
# Call the sequences
sread(d)[1:5]

##   A DNAStringSet instance of length 5
##     width seq
## [1]   100 CTGAATGCAGGTCATTTTGGTN...CGGACGNNNTCCCTGCATCCT
## [2]   100 TAGCATGCAGGAAGTCCGTTGN...CAACTCNNNAAAATTTGCCAA
## [3]   100 GCGCCTGCAGGGCTTTATCCAG...GCTGCTGNNCAGATGTCCTCC
## [4]   100 AGGTATATGCACAAAATGAGAT...CAATCTTNNCAAGCAACAGCA
## [5]   100 NCACGTGCACCAAAAAAAGAGT...NNNNNNNNNNNNNNNNNNNNN

# ... and their qualities
quality(d)[1:5]

## class: FastqQuality
## quality:
##   A BStringSet instance of length 5
##     width seq
## [1]   100 ;;<;@@2<@2222;@<<><;=#...#####################
## [2]   100 <<<??@<@=222@@@???@??#...#####################
## [3]   100 <<<@@@2<@22<@@@@?@?@@@...#####################
## [4]   100 <<<>?26@=))9>?@@@@@?<?...#####################
## [5]   100 #07>@222:))2=>@?@<;9>=...#####################
```

## Pattern matching, counting

```
grep("TATATATATATATATATATA", sread(d))

## [1]  6053 15441 82461 84384

match <- grep("TATATATATATATATATATA", sread(d))
length(match)

## [1] 4
```

## Subsetting of ShortRead object

```
d.match <- d[match]
sread(d.match)

##   A DNAStringSet instance of length 4
##     width seq
## [1]   100 TAGCATGCAGGGAGGCCTGTGT...TATTTTACACACAACGACAGA
## [2]   100 CTAGGTGCAGGTACAGTGATCG...TGCCTGCTCCCGACCGGCTTC
## [3]   100 CTGATGCAGGACAGGTCCTCCC...ATATATATATATATATATATC
## [4]   100 TAATGTGCAGGAGTCTGTAGTC...TATATATATATATATATATAT
```

## Cleaning a ShortRead object

```
d.clean <- clean(d)  # remove all reads with >= 1 'N'
sread(d.clean)[1]

##   A DNAStringSet instance of length 1
##     width seq
## [1]   100 CCATGTTGCAGGTGTGAAGGCT...GGGGACACGCCGGCCGTTTGC
```

## Trimming a ShortRead object

```
d.trim <- narrow(d.match, start = 1, end = 10)  # either end or width
quality(d.trim)

## class: FastqQuality
## quality:
##   A BStringSet instance of length 4
##     width seq
## [1]    10 ==>A<224?2
## [2]    10 BBCDF224A2
## [3]    10 @@@FFADDA2
## [4]    10 CCCFF222C2
```

Write a ShortRead object out as fastq file

```
# writeFastq(d.clean,
# file='C:/Users/daniel/Documents/science/teaching/cesky
# krumlov
# 2016/course.materials/R.files/my.clean.reads.fastq')
```

# Tasks

- Upload the stickleback data set *illumina.SE100.fastq*
- Inspect the ID, sequence and quality of the reads 1000 to 1002
- Generate a new object `X` containing the data from the reads 10001-20000
- Determine the proportion of `X`'s reads containing one or more 'N', and eliminate them from `X`
- What proportion of the filtered `X` is derived from the individual with barcode (first five bases) 'CGATA'?
- Derive the object `Y` from `X`, including only these specific CGATA-reads. Confirm that this worked by inspecting the reads
- What proportion of `Y`'s reads contains the correct restriction enzyme overhang 'TGCAGG' at the correct position (i.e., following the barcode)? Copy these reads to object `Z`
- Clip the barcodes from `Z`, then write `Z` out as a fastq file