INTRODUCTION TO BIOCONDUCTOR

Introducing ShortRead

Paula Andrea Martinez, PhD.
Data Scientist

Plant genomes

- *Arabidopsis thaliana* is a small flowering plant
- First plant to have its genome sequenced
- Genome size 135 megabase pairs (Mbp)

Sequencing companies

[1] Dan Koboldt massgenomics.org

fastq vs fasta

fastq

```
@ unique sequence identifier

raw sequence string

+ optional id

quality encoding per sequence letter
```

- fastq, fq

fasta

```
> unique sequence identifier

raw sequence string
```

- fasta, fa, seq

fasta

```
library(ShortRead)

# read fasta
fasample <- readFasta(dirPath = "data/", pattern = "fasta")

# print fasample
print(fasample)
```

```
class: ShortRead
length: 500 reads; width: 50 cycles
```

```
# methods accessors
methods(class = "ShortRead")

# Write a ShortRead object
writeFasta(fasample, file = "data/sample.fasta")
```

fastq

```
library(ShortRead)

# read fastq
fqsample <- readFastq(dirPath = "data/", pattern = "fastq")

# print fqsample
fqsample
```

```
class: ShortReadQ
length: 500 reads; width: 50 cycles
```

```
# methods accessors
methods(class = "ShortReadQ")

# Write a ShortRead object
writeFastq(fqsample, file = "data/sample.fastq.gz")
```

fastq sample

```
library(ShortRead)

# set the seed to draw the same read sequences every time
set.seed(123)

# Subsample of 500 bases
sampler <- FastqSampler("data/SRR1971253.fastq", 500)

# save the yield of 500 read sequences
sample_small <- yield(sampler)

# Class ShortReadQ
class(sample_small)

# length 500 reads
length(sample_small)
```

INTRODUCTION TO BIOCONDUCTOR

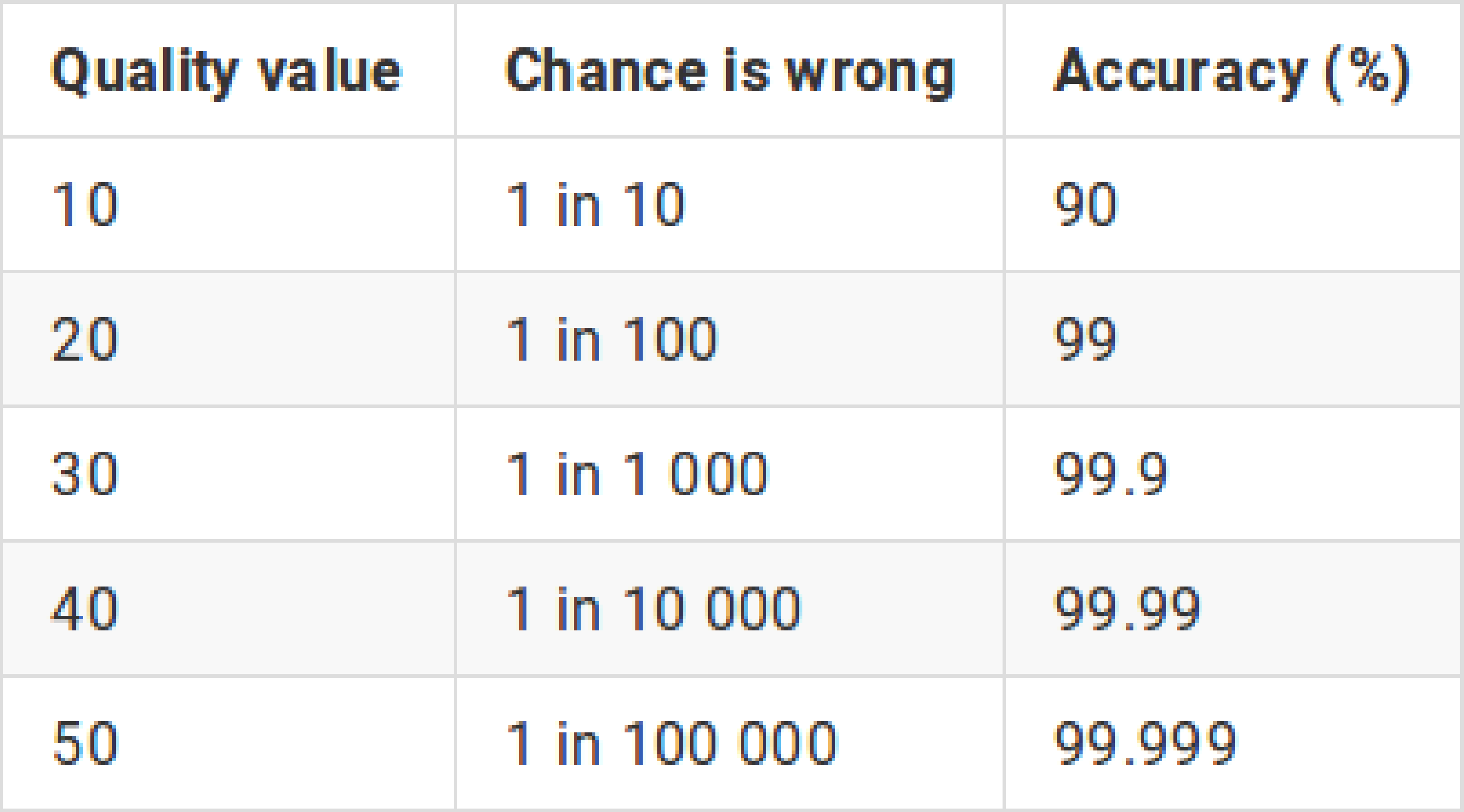
You are ready!

INTRODUCTION TO BIOCONDUCTOR

Sequence quality

Paula Andrea Martinez, PhD.
Data Scientist

Quality scores - Phred table



Quality value	Chance is wrong	Accuracy (%)
10	1 in 10	90
20	1 in 100	99
30	1 in 1 000	99.9
40	1 in 10 000	99.99
50	1 in 100 000	99.999

Encoding - Phred +33

```
# quality encoding
encoding(quality(fqsample))
```

Encoding characters and their scores

```
 !  "  #  $  %  &  '  (  )  *  +  ,  -  .   #  encoding
 0  1  2  3  4  5  6  7  8  9 10 11 12 13   #  score

 /  0  1  2  3  4  5  6  7  8  9  :  ;  <   #  encoding
14 15 16 17 18 19 20 21 22 23 24 25 26 27   #  score

 =  >  ?  @  A  B  C  D  E  F  G  H  I      #  encoding
28 29 30 31 32 33 34 35 36 37 38 39 40      #  score
```

fastq quality

```
library(ShortRead)

quality(fqsample)
```

```
class: FastqQuality
A BStringSet instance

# Quality is represented with ASCII characters
[1]     40 ?@@DDDDDHDFDHE>AHFEGFIIEBGDBHH<3FEBEEEEG
[2]     40 BCCDFFFFHHHHHJJJJJJJJJJEHHGHIJJJJJJJJJJJ
[3]     40 BCCFFFFFHFHHHJJJJJJIIJJIIIIIGIIJJIJGIJII
[4]     40 CCCFFFFFHHHHHJJJJJJJJJJIJJJJJJJJJJJJJJJJ
```

Exploring quality encoding and scores

```
library(ShortRead)

sread(fqsample)[1]
```

```
[1]    50 GTCCCATTTACCTCTGACTCTTTTGATGCTGCAATTGCTGCTCATATACT
```

```
# Quality is represented with ASCII characters
quality(fqsample)[1]
```

```
[1]    50 ?@@DDDDDHDFDHE>AHFEGFIIEBGDBHH<3FEBEEEEGGIGIIGHGHC
```

```
## PhredQuality instance
pq <- PhredQuality(quality(fqsample))

# transform encoding into scores
qs <- as(pq, "IntegerList")
qs # print scores
```

```
30 31 31 35 35 35 35 35 39 35 37 35 39 36 29 32 39 37 36 38
37 40 40 36 33 38 35 33 39 39 27 18 37 36 33 36 36 36 36 38
38 40 38 40 40 38 39 38 39 34
```

Quality assessment

```
library(ShortRead)

# Quality assessment
qaSummary <- qa(fqsample, lane = 1)    # optional lane
```

```
# class: ShortReadQQA(10)
# Names accessible with the quality assessment summary
names(qaSummary)
```

```
 [1] "readCounts"           "baseCalls"            "readQualityScore"
 [4] "baseQuality"          "alignQuality"         "frequentSequences"
 [7] "sequenceDistribution" "perCycle"             "perTile"
[10] "adapterContamination"
# QA elements are accessed with qa[["name"]]
```

```
# Get a HTML report
browseURL(report(qaSummary))
```

Alphabet by cycle

```
library(ShortRead)

# sequences alphabet
alphabet(sread(fullSample))
# [1] A,C,G,T,M,R,W,S,Y,K,V,H,D,B,N,-,+,.

abc <- alphabetByCycle(sread(fullSample))

# each observation is a letter and each variable is a cycle
# first, select the four first rows nucleotides A, C, G, T
# then, transpose
nucByCycle <- t(abc[1:4,])

nucByCycle <- nucByCycle %>%
  as.tibble() %>% # convert to tibble
  mutate(cycle = 1:50) # add cycle numbers

nucByCycle
```

```
    A     C     G     T cycle
16839 16335 16740 10878     1
13056 13327 12064 22389     2
13666 15617 13198 18355     3
14723 15439 14239 16435     4
```

INTRODUCTION TO BIOCONDUCTOR

Are you excited?

INTRODUCTION TO BIOCONDUCTOR

Match and filter

Paula Andrea Martinez, PhD.
Data Scientist

Duplicate sequences

- Biological sequence duplicates occur in nature
- Amplification from the steps in library preparation (PCR)
- Sequencing the sample more than once

Remove duplicates or at least mark them

- Whole genome sequencing or exome sequencing

Mark duplicates using a threshold

- RNA-seq and ChIP-seq

srduplicated

```
library(ShortRead)

# Counting duplicates TRUE is the number of duplicates
table(srduplicated(dfqsample))
```

```
FALSE  TRUE
500   500
```

```
# Cleaning reads from duplicates x[fun(x)]
cleanReads <- mydReads[srduplicated(mydReads) == FALSE]

# Counting duplicates
table(srduplicated(cleanReads))
```

```
FALSE
500
```

Creating your own filters

`srFilter` to filter based on a condition `x[fun(x)]`

Filter example

```
library(ShortRead)

# Use a custom filter to remove reads from fqsample
# This filter to remove reads shorter than a min number of bases
readWidthCutOff <- srFilter(function(x) {width(x) >= minWidth},
                             name = "MinWidth")

minWidth <- 51

fqsample[readWidthCutOff(fqsample)]
```

nFilter

```
library(ShortRead)

# save your filter, .name is optional
myFilter <- nFilter(threshold = 10, .name = "cleanNFilter")

# use the filter at reading point
filtered <- readFastq(dirPath = "data",
                      pattern = ".fastq",
                       filter = myFilter)

# you will retrieve only those reads that have a maximum of 10 N's
filtered
```

idFilter and polynFilter

```
library(ShortRead)

#id filter example
myFilterID <- idFilter(regex = ":3:1")
# will return only those ids that contain the regular expression
# optional parameters are .name, fixed and exclude

# use the filter at reading point
filtered <- readFastq(dirPath = "data", pattern = ".fastq",
                       filter = myFilterID)
```

```
# filter to remove poly-A regions
myFilterPolyA <- polynFilter(threshold = 10, nuc = c("A"))
# will return the sequences that have a maximun number of 10 consecutive A's

# use the filter for subsetting
filtered[myFilterPolyA(filtered)]
```

INTRODUCTION TO BIOCONDUCTOR

Let's practice using filters!

INTRODUCTION TO BIOCONDUCTOR

Multiple and parallel sequence quality assessment

Paula Andrea Martinez, PhD.
Data Scientist

Rqc

```
library(Rqc)
```

- Uses Bioconductor packages that you have already used:
  - `Biostrings, IRanges, methods, S4vectors`
- New packages to discover in the following Bioconductor courses:
  - `Rsamtools, GenomicAlignments, GenomicFiles, BiocParallel`
- CRAN packages:
  - `Knitr, dplyr, markdown, ggplot2, digest, shiny and Rccp`

rqcQA

```
library(Rqc)

files <- # get the full path of the files you want to assess

qaRqc <- rqcQA(files)

# exploring qaRqc
class(qaRqc) # "list"
names(qaRqc) # name of the input files

# for each file
qaRqc[1]
# the class of the results is RqcResultSet
```

rqcQA arguments

```
library(Rqc)

# get the path of the files you want to assess
files <- "data/seq1.fq" "data/seq2.fq" "data/seq3.fq" "data/se4.fq"

qaRqc <- rqcQA(files, workers = 4))

# sample of sequences
set.seed(1111)

qaRqc_sample <- rqcQA(files, workers = 4, sample = TRUE, n = 500))

# paired-end files
pfiles <- "data/seq_11.fq" "data/seq1_2.fq" "data/seq2_1.fq" "data/seq2_2.fq"

qaRqc_paired <- rqcQA(pfiles, workers = 4, pair = c(1, 1, 2, 2)))
```

rqcReport and rqcResultSet

```
# create a report
reportFile <- rqcReport(qaRqc, templateFile = "myReport.Rmd")

browseURL(reportFile)

#The class of qaRqc is rqcResultSet
methods(class = "RqcResultSet")
```

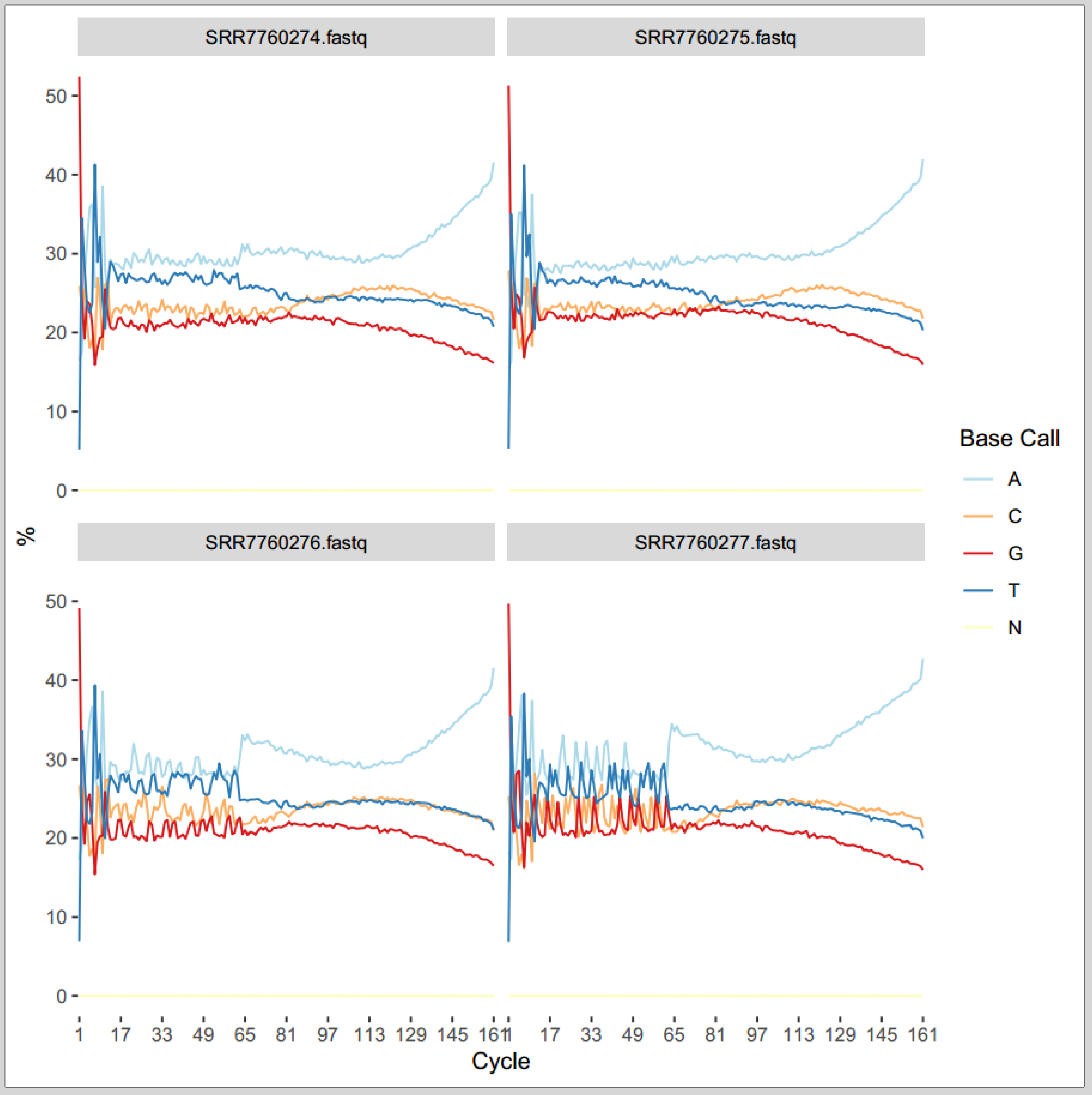
perFileInformation

```
qaRqc <- rqcQA(files, workers = 4))
perFileInformation(qaRqc)
```

filename	pair	format	group	reads	total.reads	path
SRR7760274.fastq	1	FASTQ	None	1e+06	2404795	./data
SRR7760275.fastq	2	FASTQ	None	1e+06	1508139	./data
SRR7760276.fastq	3	FASTQ	None	1e+06	1950463	./data
SRR7760277.fastq	4	FASTQ	None	1e+06	2629588	./data

Plot functions

rqc Plot functions	rqc Plot functions
`rqcCycleAverageQualityPcaPlot()`	`rqcGroupCycleAverageQualityPlot()`
`rqcCycleAverageQualityPlot()`	`rqcReadQualityBoxPlot()`
`rqcCycleBaseCallsLinePlot()`	`rqcReadQualityPlot()`
`rqcCycleBaseCallsPlot()`	`rqcReadWidthPlot()`
`rqcCycleGCPlot()`	`rqcReadFrequencyPlot()`
`rqcCycleQualityBoxPlot()`	`rqcCycleQualityPlot()`


SRR7760274.fastq
SRR7760275.fastq
SRR7760276.fastq
SRR7760277.fastq
%
50
40
30
20
10
0
1
17
33
49
65
81
97
113
129
145
161
Cycle
Base Call
A
C
G
T
N

KEEP
CALM
AND use
parallel quality
assessment

INTRODUCTION TO BIOCONDUCTOR

You are ready!

INTRODUCTION TO BIOCONDUCTOR

Congratulations!

Paula Andrea Martinez, PhD.
Data Scientist

You learned...

- Install packages from Bioconductor by using the `BiocInstaller` package.
- Techniques for reading, manipulating and filtering raw genomic data using `BioStrings`, `GenomicRanges` and `ShortRead`.
- To work with `BSgenome` and `TxDb` built-in datasets. Then used these to identify patterns by using matching functions.
- Check the quality of sequence files using `ShortRead` and `Rqc`.

You explored

INTRODUCTION TO BIOCONDUCTOR

Keep learning!