RNA-seq Analysis with *R* y Bioconductor

Antonio Miñarro and Alex Sánchez
Statistics and Bioinformatics Research Group
Departament de Estadística. Universitat de Barcelona

May 19, 2014

Contents

1 Introduction

In this lab methods for the analyis of gene expression using RNA-seq are discussed and exemplified.

1.1 R packages used in the exercise

Along the practical several packages will be used. In order to facilitate practice management they are first checked and installed if needed.

```
installifnot <- function (pkg){
  if (!require(pkg, character.only=T)){
  source("http://bioconductor.org/biocLite.R")
  biocLite(pkg)
}else{
  require(pkg, character.only=T)
  }
}
installifnot("ShortRead")
```

1.2 The IR case study

Data for the analyses are obtained from the `SRNA archives` and consist ofo four fastq files which correspond to four pools of diabetic patients (3 patients were sequenced in each pool). Pool 1 and 2 correspond to morbid obese patients without insulin resistance whereas pools 3 and 4 correspond to morbid obese patients who are insulin resistants.

The `shortRead` package contains functions to easily read fastq files (with the ReadFastq command)

```
require(ShortRead)
fq<-readFastq('Pool1small.fastq')
```

2 Data exploration and visualization

The read sequences are stored into an object of class ShortReadQ which is intended to store both sequences and quality information.

```
fq
```

```
## class: ShortReadQ
## length: 100000 reads; width: 15..72 cycles
```

```
sread(fq)
```

```
##   A DNAStringSet instance of length 100000
##           width seq
##       [1]    72 TCTTCGCCTTAATACTTTTTTATTTTGTT...AGCCTTCGTGCCCCCCCTTCCCCCTTTT
##       [2]    72 TAGAACTTGAAGGGCAAGTTGGGGGGTGN...TTAGCTCATCTAGGCTCCCCTGAAGACT
##       [3]    72 TCTCTTTAAGAGAGAGAATGTAAGGCCTN...CCTGTAATCCATACCTTTGGCAAGACCG
##       [4]    72 TGATGTGTTTTTATCCTCAAATACCTGTG...TCCTCTTAATGTCCCAAGATGAACTTGG
##       [5]    72 GGTTCAGAACGTCGTGAGACAGTTCGGTC...CGTAAGATATTTGAGAGGAGCTGTCCTT
##       ...   ... ...
##   [99996]    72 CGAACTCCTGACCTCAGGTGATCCACCTA...GTGCCAGGATTACCGGTGTGAGCCACTG
##   [99997]    72 GTTCTGTTGTCCACTAGTCGCCATCTCCA...CAAGGTTCATAAAGGGATCAAATCCCCG
##   [99998]    72 GTTCTTTTGAAAGTTTAGATAATTATTTA...TAATGGAAAAGAAAATCTGATGTTCTAT
##   [99999]    72 ATTGCTGGTGAGCTAGAGTGATTTTTGGG...TTGTTTTGTCATATTACCAGAGTTGGTT
##  [100000]    40 ATTCAGTTCTTATCCAAGAAATAACCCCGACTTAGGCTTG
```

```
sread(fq)[1:10]
```

```
##   A DNAStringSet instance of length 10
##       width seq
##   [1]    72 TCTTCGCCTTAATACTTTTTTATTTTGTTTT...TGAGCCTTCGTGCCCCCCCTTCCCCCTTTT
##   [2]    72 TAGAACTTGAAGGGCAAGTTGGGGGGTGNTC...TATTAGCTCATCTAGGCTCCCCTGAAGACT
##   [3]    72 TCTCTTTAAGAGAGAGAATGTAAGGCCTNTT...CACCTGTAATCCATACCTTTGGCAAGACCG
##   [4]    72 TGATGTGTTTTTATCCTCAAATACCTGTGAG...TCTCCTCTTAATGTCCCAAGATGAACTTGG
##   [5]    72 GGTTCAGAACGTCGTGAGACAGTTCGGTCCC...GGCGTAAGATATTTGAGAGGAGCTGTCCTT
##   [6]    72 GAATTAAGTTTACTTTCAAAAATCACTTAAA...TCAGTGAAAGGTCAGTAAAATGTAGAATTA
##   [7]    72 TAATTACATCACAAGACGTCTTGCACTCATG...CATTAGTCTTAAAAAAAGATGCAATTTCCG
##   [8]    72 ACTGTTCCCTTTGATAGAAAGGCAAAATGTC...CCGTGTTATAAATTTAACTTAATTCTAATA
##   [9]    72 AGCCATTCTACATCTTTTGATTGGAGAGTTT...ATTAAATGCTATTAGTAATGACTTACTCCT
##  [10]    47 TAGGTAAGATGTTCTTAACCCAAGCTGTTCTTTATATTCACCTGTAT
```

```
quality(fq)
```

```
## class: FastqQuality
## quality:
##   A BStringSet instance of length 100000
##           width seq
##       [1]    72 BACCBBCCBBBBBBCBBBBBBCBCBBC@B...@=?@A@@?<8>8:?><?<7<><:6:6:>
##       [2]    72 CA?BCBCA?@@375;B9.;B714'-(9(%...3'3<@><4;:A##################
##       [3]    72 BBC@CCCCC9@4A2ABBA@>:B@######...##############################
##       [4]    72 B7?B;<)?BCCCCB@@B?BBBBCA@A5@7...B?>A@BA@A<%7:5:@2'<='@@>A?##
```

```
##      [5]    72 9:;AA@2?B@9>?/;4;036</1A;0)3A...############################
##      ...   ... ...
##  [99996]    72 CCBCCBCCBBBCB@CBBB=B@@BBBBBAB...?8@@>4>=787;;<==8?7<.==8=@8;
##  [99997]    72 B;AB@B@A@?BBBB>AA>B?BA@>?@BB>...;.19:9>?9530359746<8222>9>84
##  [99998]    72 B<AB@AAA?<:>B<?@AB@AAA>>@A@@@...77:9?:;0<<?<=887?9>>9>8;>96?
##  [99999]    72 7@BBCBCB;A;@B>9@9B;@/?A?@BABB...9??:@9@@6>892=5-@<.;.859=;7=
## [100000]    40 6########################################
```

```
# width(fq)
detail(fq)
```

```
## class:  ShortReadQ
##
## sread:
##   A DNAStringSet instance of length 100000
##          width seq
##      [1]    72 TCTTCGCCTTAATACTTTTTTATTTTGTT...AGCCTTCGTGCCCCCCCTTCCCCCTTTT
##      [2]    72 TAGAACTTGAAGGGCAAGTTGGGGGGTGN...TTAGCTCATCTAGGCTCCCCTGAAGACT
##      [3]    72 TCTCTTTAAGAGAGAGAATGTAAGGCCTN...CCTGTAATCCATACCTTTGGCAAGACCG
##      [4]    72 TGATGTGTTTTTATCCTCAAATACCTGTG...TCCTCTTAATGTCCCAAGATGAACTTGG
##      [5]    72 GGTTCAGAACGTCGTGAGACAGTTCGGTC...CGTAAGATATTTGAGAGGAGCTGTCCTT
##      ...   ... ...
##  [99996]    72 CGAACTCCTGACCTCAGGTGATCCACCTA...GTGCCAGGATTACCGGTGTGAGCCACTG
##  [99997]    72 GTTCTGTTGTCCACTAGTCGCCATCTCCA...CAAGGTTCATAAAGGGATCAAATCCCCG
##  [99998]    72 GTTCTTTTGAAAGTTTAGATAATTATTTA...TAATGGAAAAGAAAATCTGATGTTCTAT
##  [99999]    72 ATTGCTGGTGAGCTAGAGTGATTTTTGGG...TTGTTTTGTCATATTACCAGAGTTGGTT
## [100000]    40 ATTCAGTTCTTATCCAAGAAATAACCCCGACTTAGGCTTG
##
## id:
##   A BStringSet instance of length 100000
##          width seq
##      [1]    28 NG-5045_Pool1_3_120_582_1069
##      [2]    24 NG-5045_Pool1_3_1_5_1101
##      [3]    23 NG-5045_Pool1_3_1_5_926
##      [4]    23 NG-5045_Pool1_3_1_5_252
##      [5]    24 NG-5045_Pool1_3_1_6_1401
##      ...   ... ...
##  [99996]    27 NG-5045_Pool1_3_5_1715_1306
##  [99997]    27 NG-5045_Pool1_3_5_1715_1580
##  [99998]    25 NG-5045_Pool1_3_5_1715_32
##  [99999]    26 NG-5045_Pool1_3_5_1715_380
## [100000]    26 NG-5045_Pool1_3_5_1715_285
## class: FastqQuality
## quality:
##   A BStringSet instance of length 100000
##          width seq
##      [1]    72 BACCBBCCBBBBBBCBBBBBBCBCBBC@B...@=?@A@@?<8>8:?><?<7<><:6:6:>
##      [2]    72 CA?BCBCA?@@375;B9.;B714'-(9(%...3'3<@><4;:A##################
##      [3]    72 BBC@CCCCC9@4A2ABBA@>:B@######...############################
##      [4]    72 B7?B;<)?BCCCCB@@B?BBBBCA@A5@7...B?>A@BA@A<%7:5:@2'<='@@>A?##
```

```
##      [5]    72 9:;AA@2?B@9>?/;4;036</1A;0)3A...############################
##      ...   ... ...
##  [99996]    72 CCBCCBCCBBBCB@CBBB=B@@BBBBBAB...?8@@>4>=787;;<==8?7<.==8=@8;
##  [99997]    72 B;AB@B@A@?BBBB>AA>B?BA@>?@BB>...;.19:9>?9530359746<8222>9>84
##  [99998]    72 B<AB@AAA?<:>B<?@AB@AAA>>@A@@@...77:9?:;0<<?<=887?9>>9>8;>96?
##  [99999]    72 7@BBCBCB;A;@B>9@9B;@/?A?@BABB...9??:@9@@6>892=5-@<.;.859=;7=
## [100000]    40 6########################################
```

2.1 Assessing sequence quality

Functions in the shortRead package can be used to check read quality.

```
qaReads<-qa(fq,lane='Pool1small')
#qaReads<-qa('.',pattern='Pool1small.fastq',type='fastq')
show(qaReads)

## class: ShortReadQQA(10)
## QA elements (access with qa[["elt"]]):
##   readCounts: data.frame(1 3)
##   baseCalls: data.frame(1 5)
##   readQualityScore: data.frame(512 4)
##   baseQuality: data.frame(94 3)
##   alignQuality: data.frame(1 3)
##   frequentSequences: data.frame(50 4)
##   sequenceDistribution: data.frame(12 4)
##   perCycle: list(2)
##     baseCall: data.frame(360 4)
##     quality: data.frame(2290 5)
##   perTile: list(2)
##     readCounts: data.frame(0 4)
##     medianReadQualityScore: data.frame(0 4)
##   adapterContamination: data.frame(1 1)

qaReads[['readCounts']] # Number of reads

##              read filter aligned
## Pool1small 100000     NA      NA

qaReads[['baseCalls']]  # Frequencies of each base

##                  A       C       G       T    N
## Pool1small 1793127 1441384 1383644 1807172 4000

head(qaReads[["frequentSequences"]], n=5)  # Frequent sequences

##                                                                  sequence
## 1 CTGTTCTTGGGTGGGTGTGGGTATAATACTAAGTTGAGATGATATCATTTACGGGGGAAGGCGCTTTGTGAA
## 2 CCCTGTTCTTGGGTGGGTGTGGGTATAATACTAAGTTGAGATGATATCATTTACGGGGGAAGGCGCTTTGTG
## 3 TTAAAGGTTCGTTTGTTCAACGATTAAAGTCCTACGTGATCTGAGTTCAGACCGGAGTAATCCAGGTCGGTT
## 4 ATTTACGGGGGAAGGCGCTTTGTGAAGTAGGCCTTATTTCTCTTGTCCTTTCGTACAGGGAGGAATTTGAAG
```

```
## 5 CTTATTTCTCTTGTCCTTTCGTACAGGGAGGAATTTGAAGTAGATAGAAACCGACCTGGATTACTCCGGTCT
##   count type       lane
## 1    14 read Pool1small
## 2    11 read Pool1small
## 3    10 read Pool1small
## 4     9 read Pool1small
## 5     9 read Pool1small
```

```
qaReads[['perCycle']]$baseCall # Base Call per cycle
```

```
##      Cycle Base Count       lane
## 1        1    A 34552 Pool1small
## 2        1    C 25140 Pool1small
## 3        1    G 18133 Pool1small
## 4        1    T 22120 Pool1small
## 15       1    N    55 Pool1small
## 19       2    A 26332 Pool1small
## 20       2    C 18808 Pool1small
## 21       2    G 20726 Pool1small
## 22       2    T 34069 Pool1small
## 33       2    N    65 Pool1small
## 37       3    A 29427 Pool1small
## 38       3    C 23003 Pool1small
## 39       3    G 16887 Pool1small
## 40       3    T 30629 Pool1small
## 51       3    N    54 Pool1small
## 55       4    A 30739 Pool1small
## 56       4    C 20424 Pool1small
## 57       4    G 17845 Pool1small
## 58       4    T 30934 Pool1small
## 69       4    N    58 Pool1small
## 73       5    A 30940 Pool1small
## 74       5    C 19629 Pool1small
## 75       5    G 21485 Pool1small
## 76       5    T 27907 Pool1small
## 87       5    N    39 Pool1small
## 91       6    A 29862 Pool1small
## 92       6    C 20306 Pool1small
## 93       6    G 20592 Pool1small
## 94       6    T 29172 Pool1small
## 105      6    N    68 Pool1small
## 109      7    A 30091 Pool1small
## 110      7    C 20200 Pool1small
## 111      7    G 21189 Pool1small
## 112      7    T 28460 Pool1small
## 123      7    N    60 Pool1small
## 127      8    A 27855 Pool1small
## 128      8    C 22227 Pool1small
## 129      8    G 18429 Pool1small
## 130      8    T 31446 Pool1small
```

```
## 141      8    N    43 Pool1small
## 145      9    A 26993 Pool1small
## 146      9    C 21484 Pool1small
## 147      9    G 19764 Pool1small
## 148      9    T 31694 Pool1small
## 159      9    N    65 Pool1small
## 163     10    A 28336 Pool1small
## 164     10    C 21351 Pool1small
## 165     10    G 20210 Pool1small
## 166     10    T 30053 Pool1small
## 177     10    N    50 Pool1small
## 181     11    A 27820 Pool1small
## 182     11    C 21712 Pool1small
## 183     11    G 20793 Pool1small
## 184     11    T 29616 Pool1small
## 195     11    N    59 Pool1small
## 199     12    A 28013 Pool1small
## 200     12    C 21898 Pool1small
## 201     12    G 21040 Pool1small
## 202     12    T 29001 Pool1small
## 213     12    N    48 Pool1small
## 217     13    A 28521 Pool1small
## 218     13    C 22024 Pool1small
## 219     13    G 20670 Pool1small
## 220     13    T 28742 Pool1small
## 231     13    N    43 Pool1small
## 235     14    A 28317 Pool1small
## 236     14    C 21988 Pool1small
## 237     14    G 21046 Pool1small
## 238     14    T 28594 Pool1small
## 249     14    N    55 Pool1small
## 253     15    A 28116 Pool1small
## 254     15    C 22216 Pool1small
## 255     15    G 20572 Pool1small
## 256     15    T 29048 Pool1small
## 267     15    N    48 Pool1small
## 271     16    A 28057 Pool1small
## 272     16    C 21691 Pool1small
## 273     16    G 20615 Pool1small
## 274     16    T 29391 Pool1small
## 285     16    N    56 Pool1small
## 289     17    A 27479 Pool1small
## 290     17    C 21964 Pool1small
## 291     17    G 20889 Pool1small
## 292     17    T 29173 Pool1small
## 303     17    N    55 Pool1small
## 307     18    A 27601 Pool1small
## 308     18    C 21617 Pool1small
## 309     18    G 20560 Pool1small
```

```
## 310     18    T 29497 Pool1small
## 321     18    N    51 Pool1small
## 325     19    A 27384 Pool1small
## 326     19    C 21266 Pool1small
## 327     19    G 21258 Pool1small
## 328     19    T 29083 Pool1small
## 339     19    N    74 Pool1small
## 343     20    A 27459 Pool1small
## 344     20    C 21674 Pool1small
## 345     20    G 20945 Pool1small
## 346     20    T 28620 Pool1small
## 357     20    N    53 Pool1small
## 361     21    A 27530 Pool1small
## 362     21    C 22060 Pool1small
## 363     21    G 20622 Pool1small
## 364     21    T 28171 Pool1small
## 375     21    N    51 Pool1small
## 379     22    A 27403 Pool1small
## 380     22    C 21557 Pool1small
## 381     22    G 21173 Pool1small
## 382     22    T 27905 Pool1small
## 393     22    N    55 Pool1small
## 397     23    A 27643 Pool1small
## 398     23    C 21564 Pool1small
## 399     23    G 20885 Pool1small
## 400     23    T 27403 Pool1small
## 411     23    N    61 Pool1small
## 415     24    A 26868 Pool1small
## 416     24    C 21884 Pool1small
## 417     24    G 21053 Pool1small
## 418     24    T 27098 Pool1small
## 429     24    N    55 Pool1small
## 433     25    A 27055 Pool1small
## 434     25    C 21726 Pool1small
## 435     25    G 20611 Pool1small
## 436     25    T 26994 Pool1small
## 447     25    N    56 Pool1small
## 451     26    A 26796 Pool1small
## 452     26    C 21129 Pool1small
## 453     26    G 20743 Pool1small
## 454     26    T 27051 Pool1small
## 465     26    N    60 Pool1small
## 469     27    A 26420 Pool1small
## 470     27    C 20859 Pool1small
## 471     27    G 21407 Pool1small
## 472     27    T 26443 Pool1small
## 483     27    N    53 Pool1small
## 487     28    A 26351 Pool1small
## 488     28    C 21041 Pool1small
```

```
## 489     28    G 21050 Pool1small
## 490     28    T 26057 Pool1small
## 501     28    N    51 Pool1small
## 505     29    A 26281 Pool1small
## 506     29    C 20937 Pool1small
## 507     29    G 20639 Pool1small
## 508     29    T 26052 Pool1small
## 519     29    N    78 Pool1small
## 523     30    A 26124 Pool1small
## 524     30    C 21051 Pool1small
## 525     30    G 20388 Pool1small
## 526     30    T 25834 Pool1small
## 537     30    N    45 Pool1small
## 541     31    A 25949 Pool1small
## 542     31    C 20423 Pool1small
## 543     31    G 21133 Pool1small
## 544     31    T 25379 Pool1small
## 555     31    N    61 Pool1small
## 559     32    A 25460 Pool1small
## 560     32    C 20476 Pool1small
## 561     32    G 21064 Pool1small
## 562     32    T 25332 Pool1small
## 573     32    N    66 Pool1small
## 577     33    A 25724 Pool1small
## 578     33    C 20665 Pool1small
## 579     33    G 20346 Pool1small
## 580     33    T 24992 Pool1small
## 591     33    N    65 Pool1small
## 595     34    A 25540 Pool1small
## 596     34    C 20259 Pool1small
## 597     34    G 20397 Pool1small
## 598     34    T 24966 Pool1small
## 609     34    N    54 Pool1small
## 613     35    A 25262 Pool1small
## 614     35    C 20166 Pool1small
## 615     35    G 20308 Pool1small
## 616     35    T 24937 Pool1small
## 627     35    N    49 Pool1small
## 631     36    A 25224 Pool1small
## 632     36    C 19842 Pool1small
## 633     36    G 20471 Pool1small
## 634     36    T 24571 Pool1small
## 645     36    N    70 Pool1small
## 649     37    A 25013 Pool1small
## 650     37    C 20167 Pool1small
## 651     37    G 19940 Pool1small
## 652     37    T 24556 Pool1small
## 663     37    N    63 Pool1small
## 667     38    A 24572 Pool1small
```

```
## 668     38    C 20008 Pool1small
## 669     38    G 20025 Pool1small
## 670     38    T 24609 Pool1small
## 681     38    N    73 Pool1small
## 685     39    A 24601 Pool1small
## 686     39    C 19877 Pool1small
## 687     39    G 19908 Pool1small
## 688     39    T 24259 Pool1small
## 699     39    N    59 Pool1small
## 703     40    A 24482 Pool1small
## 704     40    C 19684 Pool1small
## 705     40    G 19517 Pool1small
## 706     40    T 24424 Pool1small
## 717     40    N    65 Pool1small
## 721     41    A 24421 Pool1small
## 722     41    C 19545 Pool1small
## 723     41    G 19571 Pool1small
## 724     41    T 24022 Pool1small
## 735     41    N    50 Pool1small
## 739     42    A 23931 Pool1small
## 740     42    C 19810 Pool1small
## 741     42    G 19640 Pool1small
## 742     42    T 23669 Pool1small
## 753     42    N    65 Pool1small
## 757     43    A 23813 Pool1small
## 758     43    C 19745 Pool1small
## 759     43    G 19259 Pool1small
## 760     43    T 23741 Pool1small
## 771     43    N    52 Pool1small
## 775     44    A 23862 Pool1small
## 776     44    C 19504 Pool1small
## 777     44    G 19168 Pool1small
## 778     44    T 23566 Pool1small
## 789     44    N    52 Pool1small
## 793     45    A 23649 Pool1small
## 794     45    C 19550 Pool1small
## 795     45    G 18841 Pool1small
## 796     45    T 23588 Pool1small
## 807     45    N    58 Pool1small
## 811     46    A 23708 Pool1small
## 812     46    C 19240 Pool1small
## 813     46    G 18937 Pool1small
## 814     46    T 23304 Pool1small
## 825     46    N    53 Pool1small
## 829     47    A 22941 Pool1small
## 830     47    C 18854 Pool1small
## 831     47    G 18337 Pool1small
## 832     47    T 23006 Pool1small
## 843     47    N    52 Pool1small
```

```
## 847     48    A 22827 Pool1small
## 848     48    C 18895 Pool1small
## 849     48    G 18198 Pool1small
## 850     48    T 22699 Pool1small
## 861     48    N    55 Pool1small
## 865     49    A 22652 Pool1small
## 866     49    C 18945 Pool1small
## 867     49    G 17954 Pool1small
## 868     49    T 22566 Pool1small
## 879     49    N    58 Pool1small
## 883     50    A 22475 Pool1small
## 884     50    C 19101 Pool1small
## 885     50    G 17731 Pool1small
## 886     50    T 22402 Pool1small
## 897     50    N    50 Pool1small
## 901     51    A 21927 Pool1small
## 902     51    C 19071 Pool1small
## 903     51    G 17779 Pool1small
## 904     51    T 22442 Pool1small
## 915     51    N    52 Pool1small
## 919     52    A 22246 Pool1small
## 920     52    C 18544 Pool1small
## 921     52    G 17853 Pool1small
## 922     52    T 22085 Pool1small
## 933     52    N    60 Pool1small
## 937     53    A 21888 Pool1small
## 938     53    C 18822 Pool1small
## 939     53    G 17582 Pool1small
## 940     53    T 21981 Pool1small
## 951     53    N    53 Pool1small
## 955     54    A 22053 Pool1small
## 956     54    C 18303 Pool1small
## 957     54    G 17510 Pool1small
## 958     54    T 21938 Pool1small
## 969     54    N    56 Pool1small
## 973     55    A 21582 Pool1small
## 974     55    C 18521 Pool1small
## 975     55    G 17597 Pool1small
## 976     55    T 21738 Pool1small
## 987     55    N    54 Pool1small
## 991     56    A 21511 Pool1small
## 992     56    C 18309 Pool1small
## 993     56    G 17288 Pool1small
## 994     56    T 21976 Pool1small
## 1005    56    N    55 Pool1small
## 1009    57    A 21061 Pool1small
## 1010    57    C 18519 Pool1small
## 1011    57    G 17309 Pool1small
## 1012    57    T 21885 Pool1small
```

```
## 1023    57    N    40 Pool1small
## 1027    58    A 21233 Pool1small
## 1028    58    C 18252 Pool1small
## 1029    58    G 17127 Pool1small
## 1030    58    T 21764 Pool1small
## 1041    58    N    63 Pool1small
## 1045    59    A 20832 Pool1small
## 1046    59    C 18246 Pool1small
## 1047    59    G 17292 Pool1small
## 1048    59    T 21652 Pool1small
## 1059    59    N    61 Pool1small
## 1063    60    A 20539 Pool1small
## 1064    60    C 18486 Pool1small
## 1065    60    G 17254 Pool1small
## 1066    60    T 21282 Pool1small
## 1077    60    N    71 Pool1small
## 1081    61    A 20930 Pool1small
## 1082    61    C 18154 Pool1small
## 1083    61    G 17207 Pool1small
## 1084    61    T 21158 Pool1small
## 1095    61    N    37 Pool1small
## 1099    62    A 21030 Pool1small
## 1100    62    C 17801 Pool1small
## 1101    62    G 17192 Pool1small
## 1102    62    T 21214 Pool1small
## 1113    62    N    40 Pool1small
## 1117    63    A 20518 Pool1small
## 1118    63    C 18513 Pool1small
## 1119    63    G 17027 Pool1small
## 1120    63    T 20924 Pool1small
## 1131    63    N    60 Pool1small
## 1135    64    A 20721 Pool1small
## 1136    64    C 18066 Pool1small
## 1137    64    G 16961 Pool1small
## 1138    64    T 21242 Pool1small
## 1149    64    N    52 Pool1small
## 1153    65    A 20628 Pool1small
## 1154    65    C 18209 Pool1small
## 1155    65    G 17026 Pool1small
## 1156    65    T 21126 Pool1small
## 1167    65    N    53 Pool1small
## 1171    66    A 20862 Pool1small
## 1172    66    C 18057 Pool1small
## 1173    66    G 17008 Pool1small
## 1174    66    T 21067 Pool1small
## 1185    66    N    48 Pool1small
## 1189    67    A 20752 Pool1small
## 1190    67    C 17920 Pool1small
## 1191    67    G 17301 Pool1small
```

```
## 1192    67    T 21011 Pool1small
## 1203    67    N    58 Pool1small
## 1207    68    A 20850 Pool1small
## 1208    68    C 17887 Pool1small
## 1209    68    G 17273 Pool1small
## 1210    68    T 20974 Pool1small
## 1221    68    N    58 Pool1small
## 1225    69    A 20855 Pool1small
## 1226    69    C 17966 Pool1small
## 1227    69    G 17161 Pool1small
## 1228    69    T 21003 Pool1small
## 1239    69    N    57 Pool1small
## 1243    70    A 20784 Pool1small
## 1244    70    C 18100 Pool1small
## 1245    70    G 17278 Pool1small
## 1246    70    T 20837 Pool1small
## 1257    70    N    43 Pool1small
## 1261    71    A 21019 Pool1small
## 1262    71    C 17978 Pool1small
## 1263    71    G 17163 Pool1small
## 1264    71    T 20836 Pool1small
## 1275    71    N    46 Pool1small
## 1279    72    A 20835 Pool1small
## 1280    72    C 18474 Pool1small
## 1281    72    G 17492 Pool1small
## 1282    72    T 20192 Pool1small
## 1293    72    N    49 Pool1small
```

```
qaReads[['perCycle']]$quality[1:50,]  # Quality per cycle
```

```
##     Cycle Quality Score Count       lane
## 4       1       #     2  7916 Pool1small
## 6       1       %     4   135 Pool1small
## 7       1       &     5    22 Pool1small
## 8       1       '     6   157 Pool1small
## 9       1       (     7   171 Pool1small
## 10      1       )     8   162 Pool1small
## 11      1       *     9   120 Pool1small
## 12      1       +    10   173 Pool1small
## 13      1       ,    11   105 Pool1small
## 14      1       -    12   187 Pool1small
## 15      1       .    13   143 Pool1small
## 16      1       /    14   142 Pool1small
## 17      1       0    15   165 Pool1small
## 18      1       1    16   240 Pool1small
## 19      1       2    17   275 Pool1small
## 20      1       3    18   325 Pool1small
## 21      1       4    19   421 Pool1small
## 22      1       5    20   457 Pool1small
## 23      1       6    21   495 Pool1small
```

```
## 24      1       7    22   643 Pool1small
## 25      1       8    23   547 Pool1small
## 26      1       9    24   787 Pool1small
## 27      1       :    25   794 Pool1small
## 28      1       ;    26   973 Pool1small
## 29      1       <    27  1057 Pool1small
## 30      1       =    28  1347 Pool1small
## 31      1       >    29  1590 Pool1small
## 32      1       ?    30  2409 Pool1small
## 33      1       @    31  4075 Pool1small
## 34      1       A    32  6666 Pool1small
## 35      1       B    33 34181 Pool1small
## 36      1       C    34 33120 Pool1small
## 98      2       #     2  8012 Pool1small
## 100     2       %     4   111 Pool1small
## 101     2       &     5    13 Pool1small
## 102     2       '     6    87 Pool1small
## 103     2       (     7    85 Pool1small
## 104     2       )     8    86 Pool1small
## 105     2       *     9    83 Pool1small
## 106     2       +    10   112 Pool1small
## 107     2       ,    11    65 Pool1small
## 108     2       -    12   107 Pool1small
## 109     2       .    13   100 Pool1small
## 110     2       /    14   143 Pool1small
## 111     2       0    15   154 Pool1small
## 112     2       1    16   223 Pool1small
## 113     2       2    17   232 Pool1small
## 114     2       3    18   287 Pool1small
## 115     2       4    19   369 Pool1small
## 116     2       5    20   400 Pool1small
```

```
# nucleotide distribution

axc<-alphabetByCycle(sread(fq))
axc[,1:10]
```

```
##          cycle
## alphabet  [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9] [,10]
##        A 34552 26332 29427 30739 30940 29862 30091 27855 26993 28336
##        C 25140 18808 23003 20424 19629 20306 20200 22227 21484 21351
##        G 18133 20726 16887 17845 21485 20592 21189 18429 19764 20210
##        T 22120 34069 30629 30934 27907 29172 28460 31446 31694 30053
##        M     0     0     0     0     0     0     0     0     0     0
##        R     0     0     0     0     0     0     0     0     0     0
##        W     0     0     0     0     0     0     0     0     0     0
##        S     0     0     0     0     0     0     0     0     0     0
##        Y     0     0     0     0     0     0     0     0     0     0
##        K     0     0     0     0     0     0     0     0     0     0
##        V     0     0     0     0     0     0     0     0     0     0
```

```
##         H     0     0     0     0     0     0     0     0     0     0
##         D     0     0     0     0     0     0     0     0     0     0
##         B     0     0     0     0     0     0     0     0     0     0
##         N    55    65    54    58    39    68    60    43    65    50
##         -     0     0     0     0     0     0     0     0     0     0
##         +     0     0     0     0     0     0     0     0     0     0
##         .     0     0     0     0     0     0     0     0     0     0
```

```
axc[DNA_BASES, 1:10]
```

```
##          cycle
## alphabet  [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9] [,10]
##         A 34552 26332 29427 30739 30940 29862 30091 27855 26993 28336
##         C 25140 18808 23003 20424 19629 20306 20200 22227 21484 21351
##         G 18133 20726 16887 17845 21485 20592 21189 18429 19764 20210
##         T 22120 34069 30629 30934 27907 29172 28460 31446 31694 30053
```

```
axcp <- axc/ colSums(axc)
axcp[DNA_BASES, 1:10]
```

```
##          cycle
## alphabet   [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]   [,8]   [,9]
##         A 0.3455 0.2658 0.3279 0.3867 0.3094 0.3014 0.3353 0.3504 0.2699
##         C 0.2514 0.1905 0.2576 0.2581 0.1963 0.2056 0.2262 0.2809 0.2148
##         G 0.1813 0.2106 0.1904 0.2264 0.2149 0.2092 0.2389 0.2338 0.1976
##         T 0.2212 0.3473 0.3474 0.3944 0.2791 0.2974 0.3228 0.4009 0.3169
##          cycle
## alphabet  [,10]
##         A 0.2860
##         C 0.2162
##         G 0.2053
##         T 0.3064
```

```
axc2<-prop.table(axc, margin=2)*100
```

```
axc2<-axc2[DNA_BASES,]
```

```
tables(fq)
```

```
## $top
## CTGTTCTTGGGTGGGTGTGGGTATAATACTAAGTTGAGATGATATCATTTACGGGGGAAGGCGCTTTGTGAA
##                                                                       14
## CCCTGTTCTTGGGTGGGTGTGGGTATAATACTAAGTTGAGATGATATCATTTACGGGGGAAGGCGCTTTGTG
##                                                                       11
## TTAAAGGTTCGTTTGTTCAACGATTAAAGTCCTACGTGATCTGAGTTCAGACCGGAGTAATCCAGGTCGGTT
##                                                                       10
## ATTTACGGGGGAAGGCGCTTTGTGAAGTAGGCCTTATTTCTCTTGTCCTTTCGTACAGGGAGGAATTTGAAG
##                                                                        9
## CTTATTTCTCTTGTCCTTTCGTACAGGGAGGAATTTGAAGTAGATAGAAACCGACCTGGATTACTCCGGTCT
##                                                                        9
## TTTAAGACCCTCATCAATAGATGGAGACATACAGAAATAGTCAAACCACATCTACAAAATGCCAGTATCAGG
```

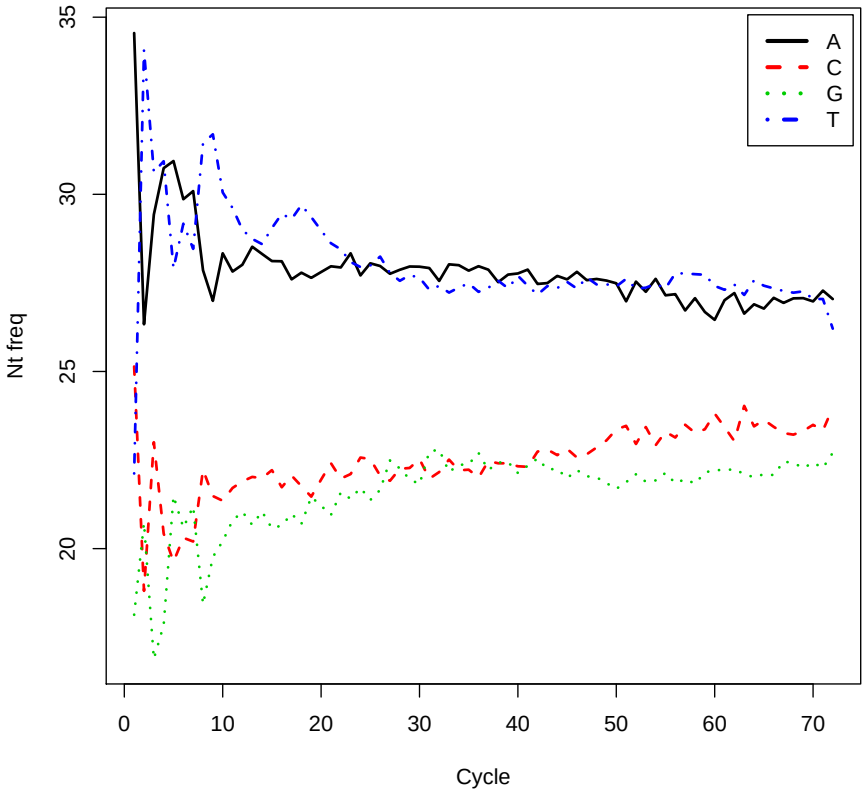
```
##                                                                      9
## ACTAAGTTGAGATGATATCATTTACGGGGGAAGGCGCTTTGTGAAGTAGGCCTTATTTCTCTTGTCCTTTCG
##                                                                      8
## ATTTCTCTTGTCCTTTCGTACAGGGAGGAATTTGAAGTAGATAGAAACCGACCTGGATTACTCCGGTCTGAA
##                                                                      8
## ACTCAATTGATCCAATAACTTGACCAACGGAACAAGTTACCCTAGGGATAACAGCGCAATCCTATTCTAGAG
##                                                                      7
## ATGCTAAGACTTCACCAGTCAAAGCGAACTACTATACTCAATTGATCCAATAACTTGACCAACGGAACAAGT
##                                                                      7
## CCAACATCGAGGTCGTAAACCCTATTGTTGATATGGACTCTAGAATAGGATTGCGCTGTTATCCCTAGGGTA
##                                                                      7
## CTTGTCCTTTCGTACAGGGAGGAATTTGAAGTAGATAGAAACCGACCTGGATTACTCCGGTCTGAACTCAGA
##                                                                      7
## GTTATCCCTAGGGTAACTTGTTCCGTTGGTCAAGTTATTGGATCAATTGAGTATAGTAGTTCGCTTTGACTG
##                                                                      7
## TGTTGGATCAGGACATCCCAATGGTGCAGCCGCTATTAAAGGTTCGTTTGTTCAACGATTAAAGTCCTACGT
##                                                                      7
## TTATTTCTCTTGTCCTTTCGTACAGGGAGGAATTTGAAGTAGATAGAAACCGACCTGGATTACTCCGGTCTG
##                                                                      7
## ACAAACCCTGTTCTTGGGTGGGTGTGGGTATAATACTAAGTTGAGATGATATCATTTACGGGGGAAGGCGCT
##                                                                      6
## ATCAATAGATGGAGACATACAGAAATAGTCAAACCACATCTACAAAATGCCAGTATCAGGCGGCGGCTTCGA
##                                                                      6
## ATTGATCCAATAACTTGACCAACGGAACAAGTTACCCTAGGGATAACAGCGCAATCCTATTCTAGAGTCCAT
##                                                                      6
## CATAGGGTCTTCTCGTCTTGCTGTGTTATGCCCGCCTCTTCACGGGCAGGTCAATTTCACTGGTTAAAAGTA
##                                                                      6
## CCCTCATCAATAGATGGAGACATACAGAAATAGTCAAACCACATCTACAAAATGCCAGTATCAGGCGGCGGC
##                                                                      6
## GATATCATTTACGGGGGAAGGCGCTTTGTGAAGTAGGCCTTATTTCTCTTGTCCTTTCGTACAGGGAGGAAT
##                                                                      6
## TTTTATGTGTTGTCGTGCAGGTAGAGGCTTACTAGAAGTGTGAAAACGTAGGCTTGGATTAAGGCGACAGCG
##                                                                      6
## TTTTTACAAACCCTTGTGTCGAGGGCTGACTTTCAATAGATCGCAGCGAGGGAGCTGCTCTGCTACGTACGA
##                                                                      6
## AAACCCTGTTCTTGGGTGGGTGTGGGTATAATACTAAGTTGAGATGATATCATTTACGGGGGAAGGCGCTTT
##                                                                      5
##                            AAGACTATACTTTCAGGGATCATTTCTATAGTGTGTTACTAGAGAA
##                                                                      5
## AGAGAAATAAGGCCTACTTCACAAAGCGCCTTCCCCCCGTAAATGATATCATCTCAACTTAGTATTATACCCA
##                                                                      5
## AGTATAGTAGTTCGCTTTGACTGGTGAAGTCTTAGCATGTACTGCTCGGAGGTTGGGTTCTGCTCCGAGGTC
##                                                                      5
## ATACAGAAATAGTCAAACCACATCTACAAAATGCCAGTATCAGGCGGCGGCTTCGAAGCCAAAGTGATGTTT
##                                                                      5
## ATACTAAGTTGAGATGATATCATTTACGGGGGAAGGCGCTTTGTGAAGTAGGCCTTATTTCTCTTGTCCTTT
##                                                                      5
## ATATAGTCACTCCAGGTTTATGGAGGGTTCTTCTACTATTAGGACTTTTCGCTTCGAAGCGAAGGCTTCTCA
##                                                                      5
```

```
## ATCCAATAACTTGACCAACGGAACAAGTTACCCTAGGGATAACAGCGCAATCCTATTCTAGAGTCCATATCA
##                                                                        5
## ATCCCTAGGGTAACTTGTTCCGTTGGTCAAGTTATTGGATCAATTGAGTATAGTAGTTCGCTTTGACTGGTG
##                                                                        5
## ATCTACTTCAAATTCCTCCCTGTACGAAAGGACAAGAGAAATAAGGCCTACTTCACAAAGCGCCTTCCCCCG
##                                                                        5
## ATGTGTCCTGCAATTCACATTAATTCTCGCAGCTAGCTGCGTTCTTCATCGACGCACGAGCCGAGTGATCCA
##                                                                        5
## CAACGGAACAAGTTACCCTAGGGATAACAGCGCAATCCTATTCTAGAGTCCATATCAACAATAGGGTTTACG
##                                                                        5
## CAATTGATCCAATAACTTGACCAACGGAACAAGTTACCCTAGGGATAACAGCGCAATCCTATTCTAGAGTCC
##                                                                        5
## CAGTCAAAGCGAACTACTATACTCAATTGATCCAATAACTTGACCAACGGAACAAGTTACCCTAGGGATAAC
##                                                                        5
## CATAATATTTCGCCCACTAAGCCAATCACTTTATTGACTCCTAGCCGCAGACCTCCTCATTCTAACCTGAGT
##                                                                        5
## CCTATTGTTGATATGGACTCTAGAATAGGATTGCGCTGTTATCCCTAGGGTAACTTGTTCCGTTGGTCAAGT
##                                                                        5
## CGGAACAAGTTACCCTAGGGATAACAGCGCAATCCTATTCTAGAGTCCATATCAACAATAGGGTTTACGACC
##                                                                        5
## CTAAGTTGAGATGATATCATTTACGGGGGAAGGCGCTTTGTGAAGTAGGCCTTATTTCTCTTGTCCTTTCGT
##                                                                        5
## CTACTATACTCAATTGATCCAATAACTTGACCAACGGAACAAGTTACCCTAGGGATAACAGCGCAATCCTAT
##                                                                        5
## CTCAACATTTTTTGTAGCCACAGGCTTCCACGGACTTCACGTCATTATTGGCTCAACTTTCCTCACTATCTG
##                                                                        5
## CTCATCAATAGATGGAGACATACAGAAATAGTCAAACCACATCTACAAAATGCCAGTATCAGGCGGCGGCTT
##                                                                        5
## CTGTTATCCCTAGGGTAACTTGTTCCGTTGGTCAAGTTATTGGATCAATTGAGTATAGTAGTTCGCTTTGAC
##                                                                        5
## CTTTCGTCACCCATGCAACAGGGTGTTTCAGTTTCATGACAAAGAATAAGGAGTCCAGAGCTTGCATGTTAA
##                                                                        5
## GATTAAAGTCCTACGTGATCTGAGTTCAGACCGGAGTAATCCAGGTCGGTTTCTATCTACTTCAAATTCCTC
##                                                                        5
## GCTAAGACTTCACCAGTCAAAGCGAACTACTATACTCAATTGATCCAATAACTTGACCAACGGAACAAGTTA
##                                                                        5
## GGGTATAATACTAAGTTGAGATGATATCATTTACGGGGGAAGGCGCTTTGTGAAGTAGGCCTTATTTCTCTT
##                                                                        5
## TAAAGTCCTACGTGATCTGAGTTCAGACCGGAGTAATCCAGGTCGGTTTCTATCTACTTCAAATTCCTCCCT
##                                                                        5
##
## $distribution
##    nOccurrences nReads
## 1             1  95088
## 2             2   1735
## 3             3    251
## 4             4     91
## 5             5     30
## 6             6      8
```

```
## 7             7      7
## 8             8      2
## 9             9      3
## 10           10      1
## 11           11      1
## 12           14      1

# nucleotide with maximum frequency per cycle and overrepresented sequence

paste(DNA_BASES[apply(axc, 2, which.max)],
      collapse="")

## [1] "ATTTAAATTTTTTTTTTTTTTTTATATTAAAAAAAAAATAAAAAAAATAAATATATTTTTTTTTTTTTTTTAA"

cnt <- vcountPattern("ATTTAAA", sread(fq))
sum(cnt > 0)

## [1] 1578
```

Plots can be easily created from the values computed.

```
matplot(t(axc2), type="l", xlab="Cycle",
        ylab="Nt freq", lwd=2, lty=seq(along=DNA_BASES),
        col=seq(along=DNA_BASES))
legend("topright", DNA_BASES, lty=seq(along=DNA_BASES),
       col=seq(along=DNA_BASES), lwd=3, inset=0.01)
```



3 Data preprocessing and filtering

Quality values can be used to "clean" the dataset and produce a subset without bad quality reads.

```
# Sum of qualities per read
qsr<-alphabetScore(quality(fq))
qsr[1:10]

##  [1] 2217 1348  801 1909  972 2247 1515 2001 1558 1161

# Mean quality per read
qar<-qsr/width(fq)
qar[1:10]

##  [1] 30.79 18.72 11.12 26.51 13.50 31.21 21.04 27.79 21.64 24.70

# select reads with mean quality greater or equal to 20
fq20q<-fq[qar>=20]
length(fq20q)
```

```
## [1] 79106
```

```
axq<-alphabetByCycle(quality(fq))
axq[,1:10]
```

```
##          cycle
## alphabet  [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9] [,10]
##              0     0     0     0     0     0     0     0     0     0
##       !      0     0     0     0     0     0     0     0     0     0
##       "      0     0     0     0     0     0     0     0     0     0
##       #   7916  8012  8109  8207  8315  8404  8499  8592  8702  8793
##       $      0     0     0     0     0     0     0     0     0     0
##       %    135   111   102   114    72   104    87    91   100    86
##       &     22    13    13    10    12    17    13    13    11    12
##       '    157    87    85    78    80    69    91    76    77    96
##       (    171    85   100   109    77   105   118   126   115    96
##       )    162    86   104    98   103    85    92   101   101   104
##       *    120    83    74    63    59    81    66    71    68    69
##       +    173   112    80    79    94    96   101    87    94    97
##       ,    105    65    85    72    68    66    84    88    81    81
##       -    187   107   118   109   108    99   103   109   136   113
##       .    143   100   132   121   105    97   113   113   122   131
##       /    142   143   154   130   146   164   167   168   188   190
##       0    165   154   164   148   157   142   163   147   158   155
##       1    240   223   251   224   256   231   220   242   260   241
##       2    275   232   233   241   244   256   270   255   283   265
##       3    325   287   272   308   348   294   351   372   355   339
##       4    421   369   363   339   405   364   435   393   439   418
##       5    457   400   385   449   429   440   464   463   482   448
##       6    495   450   467   501   463   499   543   517   533   562
##       7    643   557   523   537   536   560   615   520   612   548
##       8    547   518   543   555   531   570   608   603   686   624
##       9    787   712   721   739   793   779   858   758   873   791
##       :    794   744   746   817   845   799   889   803   932   882
##       ;    973  1017  1035  1054  1055  1020  1113  1085  1145  1165
##       <   1057  1081  1050  1135  1208  1154  1241  1162  1304  1287
##       =   1347  1514  1505  1605  1623  1623  1726  1703  1798  1780
##       >   1590  1876  1825  1958  2066  2080  2162  2195  2201  2231
##       ?   2409  2817  2875  3036  3157  3317  3347  3445  3536  3550
##       @   4075  5299  5298  5278  5595  5928  6023  6161  6374  6625
##       A   6666  8435  8575  8649  9101  9277  9690  9983 10042 10197
##       B  34181 35958 36652 37091 37637 38420 38388 38725 38340 39108
##       C  33120 28353 27361 26146 24312 22860 21360 20833 19852 18916
##       D      0     0     0     0     0     0     0     0     0     0
##       E      0     0     0     0     0     0     0     0     0     0
##       F      0     0     0     0     0     0     0     0     0     0
##       G      0     0     0     0     0     0     0     0     0     0
##       H      0     0     0     0     0     0     0     0     0     0
##       I      0     0     0     0     0     0     0     0     0     0
##       J      0     0     0     0     0     0     0     0     0     0
```

```
##       K      0     0     0     0     0     0     0     0     0     0
##       L      0     0     0     0     0     0     0     0     0     0
##       M      0     0     0     0     0     0     0     0     0     0
##       N      0     0     0     0     0     0     0     0     0     0
##       O      0     0     0     0     0     0     0     0     0     0
##       P      0     0     0     0     0     0     0     0     0     0
##       Q      0     0     0     0     0     0     0     0     0     0
##       R      0     0     0     0     0     0     0     0     0     0
##       S      0     0     0     0     0     0     0     0     0     0
##       T      0     0     0     0     0     0     0     0     0     0
##       U      0     0     0     0     0     0     0     0     0     0
##       V      0     0     0     0     0     0     0     0     0     0
##       W      0     0     0     0     0     0     0     0     0     0
##       X      0     0     0     0     0     0     0     0     0     0
##       Y      0     0     0     0     0     0     0     0     0     0
##       Z      0     0     0     0     0     0     0     0     0     0
##       [      0     0     0     0     0     0     0     0     0     0
##       \\     0     0     0     0     0     0     0     0     0     0
##       ]      0     0     0     0     0     0     0     0     0     0
##       ^      0     0     0     0     0     0     0     0     0     0
##       _      0     0     0     0     0     0     0     0     0     0
##       `      0     0     0     0     0     0     0     0     0     0
##       a      0     0     0     0     0     0     0     0     0     0
##       b      0     0     0     0     0     0     0     0     0     0
##       c      0     0     0     0     0     0     0     0     0     0
##       d      0     0     0     0     0     0     0     0     0     0
##       e      0     0     0     0     0     0     0     0     0     0
##       f      0     0     0     0     0     0     0     0     0     0
##       g      0     0     0     0     0     0     0     0     0     0
##       h      0     0     0     0     0     0     0     0     0     0
##       i      0     0     0     0     0     0     0     0     0     0
##       j      0     0     0     0     0     0     0     0     0     0
##       k      0     0     0     0     0     0     0     0     0     0
##       l      0     0     0     0     0     0     0     0     0     0
##       m      0     0     0     0     0     0     0     0     0     0
##       n      0     0     0     0     0     0     0     0     0     0
##       o      0     0     0     0     0     0     0     0     0     0
##       p      0     0     0     0     0     0     0     0     0     0
##       q      0     0     0     0     0     0     0     0     0     0
##       r      0     0     0     0     0     0     0     0     0     0
##       s      0     0     0     0     0     0     0     0     0     0
##       t      0     0     0     0     0     0     0     0     0     0
##       u      0     0     0     0     0     0     0     0     0     0
##       v      0     0     0     0     0     0     0     0     0     0
##       w      0     0     0     0     0     0     0     0     0     0
##       x      0     0     0     0     0     0     0     0     0     0
##       y      0     0     0     0     0     0     0     0     0     0
##       z      0     0     0     0     0     0     0     0     0     0
##       {      0     0     0     0     0     0     0     0     0     0
```

```
##       |      0     0     0     0     0     0     0     0     0     0
##       }      0     0     0     0     0     0     0     0     0     0
```

```
# select reads of 72 bp
fq72b<-fq[width(fq)==72]
length(fq72b)
```

```
## [1] 77042
```

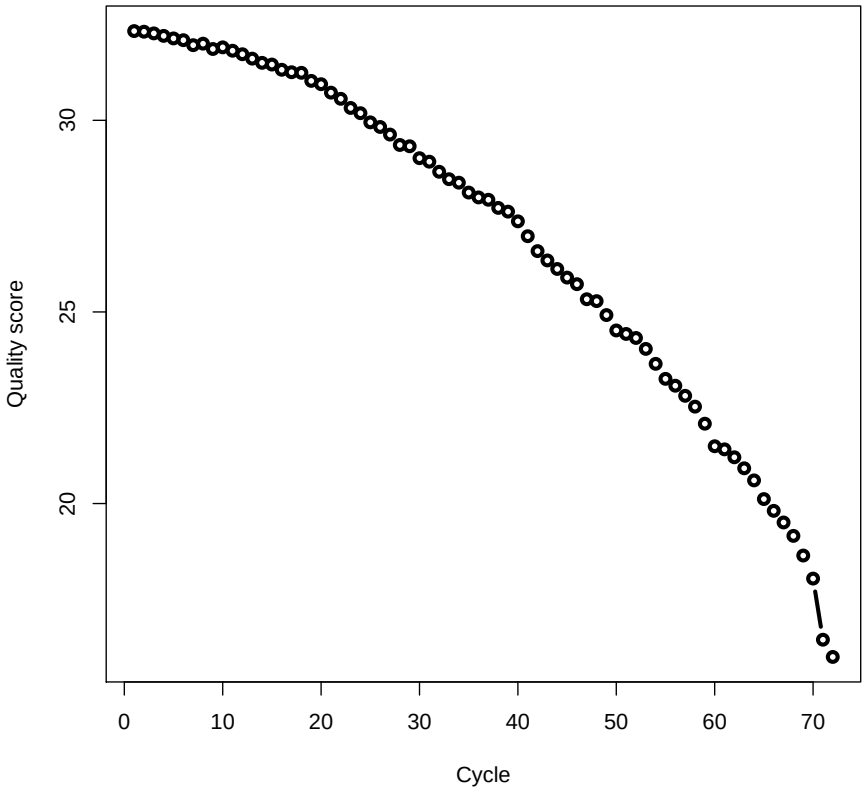
```
# mean quality per cycle
qxc <- as(quality(fq72b), "matrix")
dim(qxc)
```

```
## [1] 77042    72
```

```
qxc[1:5, 1:20]
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## [1,]   33   32   34   34   33   33   34   34   33    33    33    33    33
## [2,]   34   32   30   33   34   33   34   32   30    31    31    18    22
## [3,]   33   33   34   31   34   34   34   34   34    24    31    19    32
## [4,]   33   22   30   33   26   27    8   30   33    34    34    34    34
## [5,]   24   25   26   32   32   31   17   30   33    31    24    29    30
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20]
## [1,]    33    34    33    33    33    33    33
## [2,]    20    26    33    24    13    26    33
## [3,]    17    32    33    33    32    31    29
## [4,]    33    31    31    33    30    33    33
## [5,]    14    26    19    26    15    18    21
```

```
plot(colMeans(qxc), type="b", lwd=3, xlab="Cycle",
     ylab="Quality score")
```



Reads with "N" can be filtered out

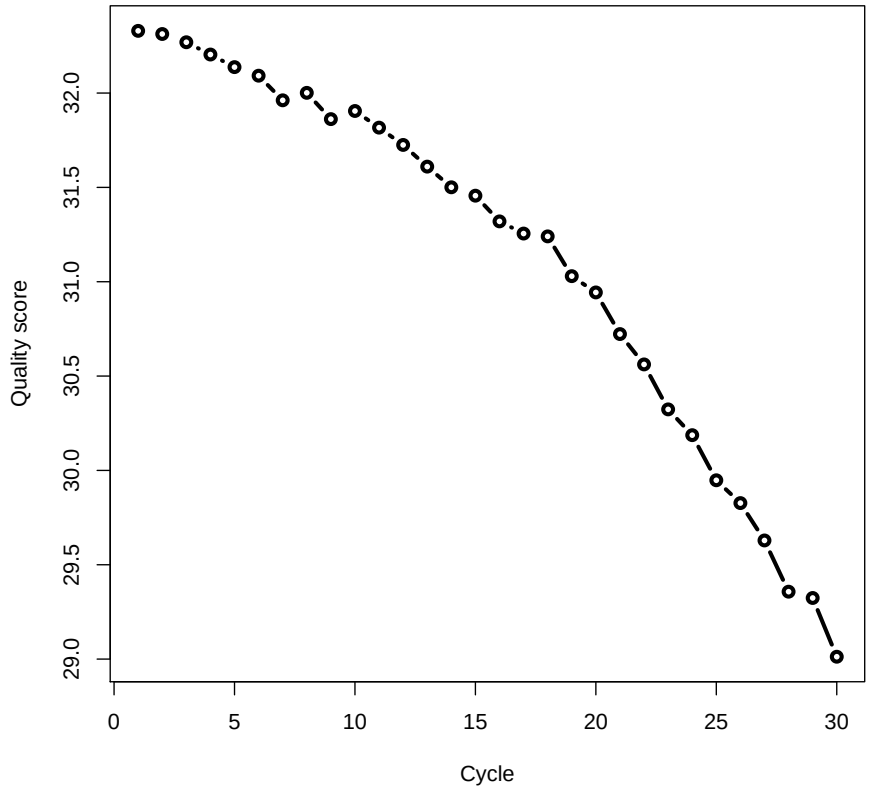
```
filt<-nFilter(threshold=0L)
fqn<-fq[filt(fq)]
length(fqn)

## [1] 96084

axc<-alphabetByCycle(sread(fqn))
axc[,1:10]

##          cycle
## alphabet  [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9] [,10]
##        A 33180 25358 28279 29582 29716 28701 28897 26772 25917 27200
##        C 24176 18131 22100 19635 18918 19573 19435 21375 20673 20523
##        G 17520 19966 16295 17172 20622 19823 20397 17754 19044 19448
##        T 21208 32629 29410 29695 26828 27987 27355 30183 30450 28913
##        M     0     0     0     0     0     0     0     0     0     0
##        R     0     0     0     0     0     0     0     0     0     0
##        W     0     0     0     0     0     0     0     0     0     0
##        S     0     0     0     0     0     0     0     0     0     0
```

```
##         Y     0     0     0     0     0     0     0     0     0     0
##         K     0     0     0     0     0     0     0     0     0     0
##         V     0     0     0     0     0     0     0     0     0     0
##         H     0     0     0     0     0     0     0     0     0     0
##         D     0     0     0     0     0     0     0     0     0     0
##         B     0     0     0     0     0     0     0     0     0     0
##         N     0     0     0     0     0     0     0     0     0     0
##         -     0     0     0     0     0     0     0     0     0     0
##         +     0     0     0     0     0     0     0     0     0     0
##         .     0     0     0     0     0     0     0     0     0     0
```

And reads containing non-nucleotide symbols can be removed

```
fqnet<-clean(fq)
length(fqnet)

## [1] 96084

axc<-alphabetByCycle(sread(fqnet))
axc[,1:10]

##          cycle
## alphabet  [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9] [,10]
##         A 33180 25358 28279 29582 29716 28701 28897 26772 25917 27200
##         C 24176 18131 22100 19635 18918 19573 19435 21375 20673 20523
##         G 17520 19966 16295 17172 20622 19823 20397 17754 19044 19448
##         T 21208 32629 29410 29695 26828 27987 27355 30183 30450 28913
##         M     0     0     0     0     0     0     0     0     0     0
##         R     0     0     0     0     0     0     0     0     0     0
##         W     0     0     0     0     0     0     0     0     0     0
##         S     0     0     0     0     0     0     0     0     0     0
##         Y     0     0     0     0     0     0     0     0     0     0
##         K     0     0     0     0     0     0     0     0     0     0
##         V     0     0     0     0     0     0     0     0     0     0
##         H     0     0     0     0     0     0     0     0     0     0
##         D     0     0     0     0     0     0     0     0     0     0
##         B     0     0     0     0     0     0     0     0     0     0
##         N     0     0     0     0     0     0     0     0     0     0
##         -     0     0     0     0     0     0     0     0     0     0
##         +     0     0     0     0     0     0     0     0     0     0
##         .     0     0     0     0     0     0     0     0     0     0

# keep only first 30 bases
fq30p<-narrow(fq72b,start=1,end=30)
sread(fq30p)

##   A DNAStringSet instance of length 77042
##          width seq
##     [1]    30 TCTTCGCCTTAATACTTTTTTATTTTGTTT
##     [2]    30 TAGAACTTGAAGGGCAAGTTGGGGGGTGNT
##     [3]    30 TCTCTTTAAGAGAGAGAATGTAAGGCCTNT
```

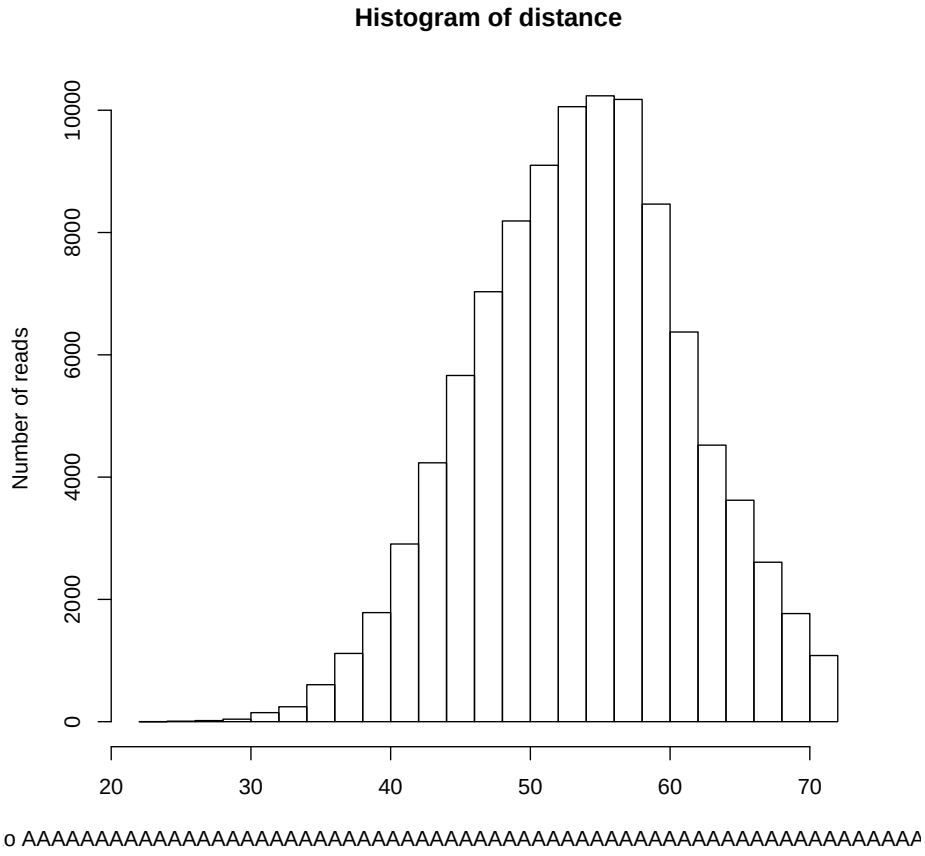
```
##      [4]    30 TGATGTGTTTTTATCCTCAAATACCTGTGA
##      [5]    30 GGTTCAGAACGTCGTGAGACAGTTCGGTCC
##      ...   ... ...
## [77038]    30 CATCTCTCAGGAAAACAGAGCTGTTGTATC
## [77039]    30 CGAACTCCTGACCTCAGGTGATCCACCTAC
## [77040]    30 GTTCTGTTGTCCACTAGTCGCCATCTCCAC
## [77041]    30 GTTCTTTTGAAAGTTTAGATAATTATTTAA
## [77042]    30 ATTGCTGGTGAGCTAGAGTGATTTTTGGGG

qxc <- as(quality(fq30p), "matrix")
plot(colMeans(qxc), type="b", lwd=3, xlab="Cycle",ylab="Quality score")
```




```
# remove sequences resembling a polyA
head(qaReads[["frequentSequences"]], n=5)  # Frequent sequences

##                                                                                                  sequence
## 1 CTGTTCTTGGGTGGGTGTGGGTATAATACTAAGTTGAGATGATATCATTTACGGGGGAAGGCGCTTTGTGAA
## 2 CCCTGTTCTTGGGTGGGTGTGGGTATAATACTAAGTTGAGATGATATCATTTACGGGGGAAGGCGCTTTGTG
## 3 TTAAAGGTTCGTTTGTTCAACGATTAAAGTCCTACGTGATCTGAGTTCAGACCGGAGTAATCCAGGTCGGTT
## 4 ATTTACGGGGGAAGGCGCTTTGTGAAGTAGGCCTTATTTCTCTTGTCCTTTCGTACAGGGAGGAATTTGAAG
```

```
## 5 CTTATTTCTCTTGTCCTTTCGTACAGGGAGGAATTTGAAGTAGATAGAAACCGACCTGGATTACTCCGGTCT
##   count type       lane
## 1    14 read Pool1small
## 2    11 read Pool1small
## 3    10 read Pool1small
## 4     9 read Pool1small
## 5     9 read Pool1small
```

```
distance<-srdistance(fq,polyn('A',width(fq)[1]))[[1]]
# histogram of distances
hist(distance,xlab=paste('Distance to',polyn('A',width(fq)[1])),ylab='Number of reads')
```




```
# create a mask to select reads that resemble the sequence
polyAs<-distance<30
head(polyAs)
```

```
## [1] FALSE FALSE FALSE FALSE FALSE FALSE
```

```
sum(polyAs)
```

```
## [1] 38
```

```
fqnoPA<-fq[!polyAs]
length(fqnoPA)

## [1] 99962
```

The clean set of sequences can be saved as a new fastq file

```
# save in fastq format
writeFastq(fqnoPA,file='fqnoPA.fastq')
```