Short Read Analysis by R

We use the pasilla data set derived from a paper [1]. The authors investigate conservation of RNA regulation between D. melanogaster and mammals. Part of their study used RNAi and RNA-seq to identify exons regulated by Pasilla (ps), the D. melanogaster ortholog of mammalian NOVA1 and NOVA2. Briefly, their experiment compared gene expression as measured by RNAseq in S2-DRSC cells cultured with, or without, a 444bp dsRNA fragment corresponding to the ps mRNA sequence. Their assessment investigated differential exon use, but our worked example will focus on gene-level differences.

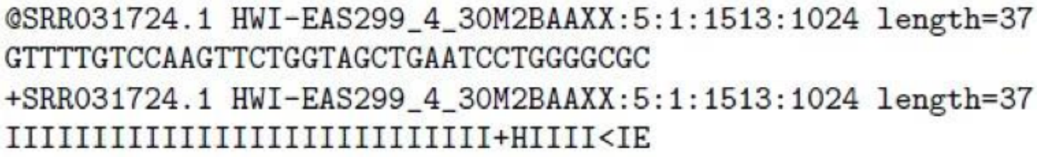
In this section we look at a subset of the ps data, corresponding to reads obtained from lanes of their RNA-seq experiment, and to the same reads aligned to a D. melanogaster reference genome. Reads were obtained from GEO and the Short Read Archive (SRA), and were aligned to the D. melanogaster reference genome dm3 as described in the pasilla experiment data package.

Install packages and datasets

```
source("http://bioconductor.org/biocLite.R")
biocLite ("ShortRead")
library (ShortRead) #Load ShortRead package
```

Short read formats

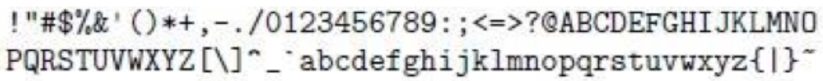
The Illumina GAII and HiSeq technologies generate sequences by measuring incorporation of florescent nucleotides over successive PCR cycles. These sequencers produce output in a variety of formats, but FASTQ is ubiquitous. Each read is represented by a record of four components:



```
@SRR031724.1 HWI-EAS299_4_30M2BAAXX:5:1:1513:1024 length=37
GTTTTGTCCAAGTTCTGGTAGCTGAATCCTGGGGCGC
+SRR031724.1 HWI-EAS299_4_30M2BAAXX:5:1:1513:1024 length=37
IIIIIIIIIIIIIIIIIIIIIIIIII+HIIII<IE
```

The first and third lines (beginning with @ and + respectively) are unique identifier. The identifier produced by the sequencer typically includes a machine id followed by colon-separated information on the lane, tile, x, and y coordinate of the read. The example illustrated here also includes the SRA accession number, added when the data was submitted to the archive. The machine identifier could potentially be used to extract information about batch effects. The spatial coordinates (lane, tile, x, y) are often used to identify optical duplicates; spatial coordinates can also be used during quality assessment to identify artifacts of sequencing, e.g., uneven amplification across the flow cell, though these spatial effects are rarely pursued.

The second and fourth lines of the FASTQ record are the nucleotides and qualities of each cycle in the read. This information is given in 5 to 3 orientation as seen by the sequencer. A letter N in the sequence is used to signify bases that the sequencer was not able to call. The fourth line of the FASTQ record encodes the quality (confidence) of the corresponding base call. The quality score is encoded following one of several conventions, with the general notion being that letters later in the visible ASCII alphabet are of lower quality; this is developed further below. Both the sequence and quality scores may span multiple lines.



```
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNO
PQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
```

Technologies other than Illumina use different formats to represent sequences. Roche 454 sequence data is generated by “flowing” labeled nucleotides over samples, with greater intensity corresponding to longer runs of A, C, G, or T. This data is represented as a series of “flow grams” (a kind of run- length encoding of the read) in Standard Flowgram Format (SFF). The Bioconductor package R453Plus1Toolbox has facilities for parsing SFF files, but after quality control steps the data are frequently represented (with some loss of information) as FASTQ. SOLiD technologies produce sequence data using a “color space” model. This data is not easily read in to R, and much of the error-correcting benefit of the color space model is lost when converted to FASTQ; SOLiD sequences are not well-handled by Bioconductor packages.

Read FASTQ file

FASTQ files can be read in to R using the readFastq function from the ShortRead package. Save the fastq file “SRR031724_1_subset.fastq” into your working directory.

```
fq <-readFastq("SRR031724_1_subset.fastq") #Read a FASTQ file
```

```
> fq
class: ShortReadQ
length: 1000000 reads; width: 37 cycles
```

The data are represented as an object of class ShortReadQ.

```
> head(sread(fq), 3)
  A DNAStringSet instance of length 3
    width seq
[1]    37 GTTTTGTCCAAGTTCTGGTAGCTGAATCCTGGGGCGC
[2]    37 GTTGTCGCATTCCTTACTCTCATTCGGGAATTCTGTT
[3]    37 GAATTTTTTGAGAGCGAAATGATAGCCGATGCCCTGA


> head(quality(fq), 3)
class: FastqQuality
quality:
  A BStringSet instance of length 3
    width seq
[1]    37 IIIIIIIIIIIIIIIIIIIIIIIIIIII+HIIII<IE
[2]    37 IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
[3]    37 IIIIIIIIIIIIIIIIIIIIII'IIIIIGBIIII2I+

> head(id(fq), 3)
  A BStringSet instance of length 3
    width seq
[1]    58 SRR031724.1 HWI-EAS299_4_30M2BAAXX:5:1:1513:1024 length=37
[2]    57 SRR031724.2 HWI-EAS299_4_30M2BAAXX:5:1:937:1157 length=37
[3]    58 SRR031724.4 HWI-EAS299_4_30M2BAAXX:5:1:1443:1122 length=37
```

The ShortReadQ class illustrates class inheritance. It extends the ShortRead class

```
> getClass("ShortReadQ")
Class "ShortReadQ" [package "ShortRead"]

Slots:
```

```
Name:       quality        sread           id
Class: QualityScore DNAStringSet   BStringSet

Extends:
Class "ShortRead", directly
Class ".ShortReadBase", by class "ShortRead", distance 2
```

Methods defined on ShortRead are available for ShortReadQ.

```
showMethods(class="ShortRead", where=getNamespace("ShortRead"))
```

For instance, the width can be used to demonstrate that all reads consist of 37 nucleotides.

```
> table(width(fq))

     37
1000000
```

The alphabetByCycle function summarizes use of nucleotides at each cycle in a (equal width) ShortReadQ or DNAStringSet instance.

```
abc <-alphabetByCycle(sread(fq))
```

```
> abc[1:4, 1:8]
        cycle
alphabet   [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]   [,8]
       A  78194 153156 200468 230120 283083 322913 162766 220205
       C 439302 265338 362839 251434 203787 220855 253245 287010
       G 397671 270342 258739 356003 301640 247090 227811 246684
       T  84833 311164 177954 162443 211490 209142 356178 246101
```

Exercise 1
Count the number of "N"s in each cycle and graph the "N" count versus cycle number.

FASTQ files are getting larger. A very common reason for looking at data at this early stage in the processing pipeline is to explore sequence quality. In these circumstances it is often not necessary to parse the entire FASTQ file. Instead create a representative sample.

```
sampler <-FastqSampler("SRR031724_1_subset.fastq", 1000)
reads = yield(sampler) # sample of 1000 reads
```

A second common scenario is to pre-process reads, e.g., trimming low-quality tails, adapter sequences, or artifacts of sample preparation.

Exercise 2 Use quality to extract the quality scores of the short reads. Interpret the encoding qualitatively. Convert the quality scores to a numeric matrix, using as. Inspect the numeric matrix (e.g., using dim) and understand what it represents. Use colMeans to summarize the average quality score by cycle. Use plot to visualize this.

```
> head (quality (fq))
class: FastqQuality
quality:
  A BStringSet instance of length 6
    width seq
[1]    37 IIIIIIIIIIIIIIIIIIIIIIIIIIII+HIIII<IE
[2]    37 IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
[3]    37 IIIIIIIIIIIIIIIIIIIIII'IIIIIGBIIII2I+
[4]    37 IIIIIIIIIIIIIIIIIIIIIIII,II*E,&4HI++B
[5]    37 IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII&.$
[6]    37 III.IIIIIIIIIIIIIIIIIII%IIE(-EIH<IIII
```

```
qual <- as(quality(fq), "matrix")
```

Exercise 3 Trim the low quality bases in the sequences.

```
minQuality = 30 # minimum quality
firstBase = 1
minLength = 18 #minimum sequence length
qualMat<- as(quality(reads), "matrix")
qualList<-split(qualMat,row(qualMat))
ends<- lapply(qualList,function(x){which(x<minQuality)[1]-1}) ends <- as.integer(ends)
starts <- lapply(ends,function(x){min(x+1,firstBase)})
starts <- as.integer (starts)
newQ<-ShortReadQ(sread=subseq(sread(reads),start=starts,end=ends), quality=new(Class=class(quality(
    reads)),quality=subseq(quality(quality(reads)),start=starts,end=ends)),id=id(reads))

#apply minLength using srFilter
lengthCutoff <- srFilter(function(x) { width(x)>=minLength},name="length cutoff")
newQ[lengthCutoff(newQ)]
head (sread (newQ),3)
head (quality (newQ),3)
```

Reference

[1] A. N. Brooks, L. Yang, M. O. Du_, K. D. Hansen, J. W. Park, S. Dudoit,
S. E. Brenner, and B. R. Graveley. Conservation of an RNA regulatory
map between Drosophila and mammals. *Genome Research*, pages 193{202,
2011.

Note: The lab material was developed based on a tutorial by M. Carlson *et al.*: *High-throughput sequence analysis with R and Bioconductor*.