🧬 Inside Bioconductor: What I Discovered About Genomic Data with R

From DNA translation to sequence quality — decoding biology through data

Created by Md Abrar Faiyaj | M.S.in Biotechnology | R • Bioinformatics

```r
#Author: MD ABRAR FAIYAJ
#DATE: 16/10/2025

# Install BiocManager
install.packages("BiocManger")
#Install the GenomicsRanges Package
BiocManager::install("GenomicRanges")
#Load BiocManager
library(BiocManager)
#Check BIOConductor version
version()
library(GenomicRanges)
#Install BSgenome
BiocManager::install("BSgenome")
library(BSgenome)
```

Package Installation: BSgenome package is a core software package within the Bioconductor project that provides the infrastructure for efficiently storing and accessing the full DNA sequences of a specific organism's genome.

```r
# Author: MD ABRAR FAIYAJ
# DATE: 16.10.2025

# Load packages
library(Biostrings)
library(Biostrings)
# Define the file path using forward slashes
file_path <- "C:/Users/HP/Documents/data/sequence.fasta"
# Load the file
zikavirus_genome <- readRNAStringSet(file_path, format = "fasta")
# View the loaded sequence to confirm it worked
zikavirus_genome
# Create zikv with one collated sequence using zikaVirus
zikv <- unlist(zikaVirus)
# Check the length of zikaVirus and zikv
length(zikaVirus)
length(zikv)
# Complement the zikv sequence
complement(zikv)
# Reverse complement the zikv sequence
reverseComplement(zikv)
# Translate the zikv sequence
translate(zikv)
# Find palindromes in zikv
findPalindromes(zikv)
```

a) This is the code by using Biostrings package to analyze the Zika virus's genome

```r
# Reverse complement the zikv sequence
reverseComplement(zikv)
89-letter RNAString object
q: GCCUGGUUUCCCCCCCGGCCGCCGGUUCGGCGUCUGUGC...GCUCUCGCUUCGCUCGCUGUCGCGUCUGCUCCCGUC

# Translate the zikv sequence
translate(zikv)
29-letter AAString object
q: RDGSRRDSERSESNNRQQEGGNESGRKTPKKKPEDRSKA...GRGDPPENAKQQDGERPETQSHHAAARHRRRTGGRG
arning message:
 .Call2("DNAStringSet_translate", x, skip_code, dna_codes[codon_alphabet],  :
 last 2 bases were ignored


# Find palindromes in zikv
findPalindromes(zikv)
ews on a 8489-letter RNAString subject
bject: AGGGACGGGAGCAGACGCGACAGCGAGCGAAGCGAGA...ACAGACGCCGAACCGGCGGCCGGGGGGGAAACCA
ews:
     start  end width
 [1]   676  683     8 [CCCCGGGG]
 [2]   689  696     8 [GGCCGGCC]
 [3]  1340 1349    10 [CCGCGCGCGG]
 [4]  2949 2956     8 [GCCGCGGC]
 [5]  2994 3001     8 [CCGGCCGG]
 ...   ...  ...   ... ...
[19]  7799 7807     9 [GGGCAGCCC]
[20]  7838 7846     9 [GCCGGCGGC]
[21]  8172 8180     9 [CGGCAGCCG]
[22]  8440 8447     8 [GCGGCCGC]
[23]  8465 8475    11 [CCGGCGGCCGG]
```

b) This is the output result

🧬 Why Palindromic Sequences Matter in the Zika Virus's Genome ?

- **Palindromic sequences**: short regions that read the **same forward** and **backward** on complementary strands
- In viral genomes, these sequences fold into **stem-loop structures**
- Stem-loop structures stabilize **RNA** and **regulate replication efficiency**
- **Identifying palindromic regions** aids understanding of **Zika's evolution** and **mutation patterns**
- Helps in designing potential a**ntiviral targets** against **Zika virus**

🧬 **Genome Source**: NCBI Reference Sequence — NC_012532.1 (Zika virus, complete genome)

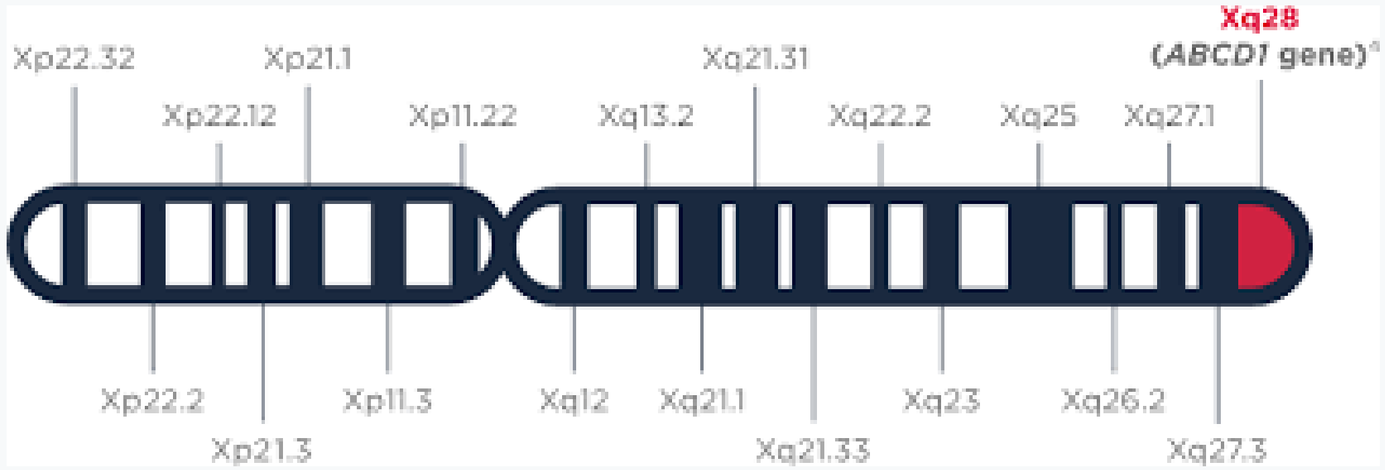
Next Step: To find out the ABCD1 Gene from the Human X-Chromosome

- Gene of Interest: **ABCD1**

- ABCD1 is located at the end of the chromosome X long arm

- It encodes a protein relevant for the well functioning of brain and lung cells in mammals

- Chr X is almost ~ 156 mi bp long
- The ABCD1 gene is located in the X chromosome at a position known as Xq28.



Source: https://www.itmightbeald.com/understanding-aldt

STEP 1: Extract all positive stranded genes in chromosome X

```r
#Author: MD ABRAR FAIYAJ
#DATE: 16/10/2025

# Load human reference genome hg38
library(TxDb.Hsapiens.UCSC.hg38.knownGene)

# Assign hg38 to hg, then print it
hg <- TxDb.Hsapiens.UCSC.hg38.knownGene
hg

# Extract all positive stranded genes in chromosome X, assign to hg_chrXgp, then sort it
hg_chrXgp <- genes(hg, filter = list(tx_chrom = "chrX", tx_strand = "+"))
sort(hg_chrXgp)
```

Here, we can find all positive strand genes in Chromosome X by using this code

Result of the Step-1

```
> # Assign hg38 to hg, then print it
> hg <- TxDb.Hsapiens.UCSC.hg38.knownGene
> hg
TxDb object:
# Db type: TxDb
# Supporting package: GenomicFeatures
# Data source: UCSC
# Genome: hg38
# Organism: Homo sapiens
# Taxonomy ID: 9606
# UCSC Table: knownGene
# UCSC Track: GENCODE V47
# Resource URL: https://genome.ucsc.edu/
# Type of Gene ID: Entrez Gene ID
# Full dataset: yes
# miRBase build ID: NA
# Nb of transcripts: 412034
# Db created by: txdbmaker package from Bioconductor
# Creation time: 2025-03-02 02:45:03 +0000 (Sun, 02 Mar 2025)
# txdbmaker version at creation time: 1.3.1
# RSQLite version at creation time: 2.3.9
# DBSCHEMAVERSION: 1.2
>
> # Extract all positive stranded genes in chromosome X, assign to hg_chrXgp, then sort it
> hg_chrXgp <- genes(hg, filter = list(tx_chrom = "chrX", tx_strand = "+"))
> sort(hg_chrXgp)
GRanges object with 686 ranges and 1 metadata column:
            seqnames              ranges strand |     gene_id
               <Rle>           <IRanges>  <Rle> | <character>
      55344     chrX       276322-303356      + |       55344
  102724521     chrX       386983-511616      + |   102724521
       6473     chrX       624344-659411      + |        6473
       1438     chrX     1268793-1325373      + |        1438
  100500894     chrX     1293918-1293992      + |   100500894
        ...      ...                 ...    ... .         ...
  100422977     chrX 155457517-155457615      + |   100422977
  100507404     chrX 155457738-155611616      + |   100507404
      10251     chrX 155612572-155782459      + |       10251
       6845     chrX 155881345-155943769      + |        6845
       3581     chrX 155997696-156022236      + |        3581
  -------
  seqinfo: 711 sequences (1 circular) from hg38 genome
>
```

Here, we see that the output of the **step-1**

Step-2: To find out the **ABCD1 gene**

```
> hg_chrXt <- transcriptsBy(hg, by = "gene")
> hg_chrXt
GRangesList object of length 1271:
$`100008586`
GRanges object with 2 ranges and 2 metadata columns:
      seqnames            ranges strand |     tx_id           tx_name
         <Rle>         <IRanges>  <Rle> | <integer>       <character>
  [1]     chrX 49551278-49568218      + |    374347 ENST00000639028.1
  [2]     chrX 49560842-49568205      + |    374349 ENST00000440137.2
  -------
  seqinfo: 1 sequence from hg38 genome

$`10009`
GRanges object with 2 ranges and 2 metadata columns:
      seqnames              ranges strand |     tx_id           tx_name
         <Rle>           <IRanges>  <Rle> | <integer>       <character>
  [1]     chrX 120250752-120258398      + |    376650 ENST00000326624.2
  [2]     chrX 120250812-120258398      + |    376651 ENST00000557385.2
  -------
  seqinfo: 1 sequence from hg38 genome
                                                t
$`100093698`
GRanges object with 1 range and 2 metadata columns:
      seqnames            ranges strand |     tx_id           tx_name
         <Rle>         <IRanges>  <Rle> | <integer>       <character>
  [1]     chrX 13310652-13319933      + |    372957 ENST00000431486.1
  -------
  seqinfo: 1 sequence from hg38 genome

...
<1268 more elements>
>
> # Select gene `215` from the hg_chrXt
> hg_chrXt$'215'
GRanges object with 3 ranges and 2 metadata columns:
      seqnames              ranges strand |     tx_id           tx_name
         <Rle>           <IRanges>  <Rle> | <integer>       <character>
  [1]     chrX 153724856-153744755      + |    377797 ENST00000218104.6
  [2]     chrX 153725817-153729897      + |    377798 ENST00000370129.4
  [3]     chrX 153735344-153740604      + |    377799 ENST00000443684.2
  -------
  seqinfo: 1 sequence from hg38 genome
>
```

Here,The gene id of **ABCD1 is "215"** and we find the gene of interest

Introducing the ShortRead Package

```r
#Author: MD ABRAR FAIYAJ
#DATE: 16/10/2025

# Exploring sequence quality

# load ShortRead
library(ShortRead)

# Check quality
quality(fqsample)

# Check encoding of quality
encoding(quality(fqsample))

# Check baseQuality
qaSummary[["baseQuality"]]

# very important for visualization
browseURL(report(qaSummary))
```

Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

ShortRead Quality Assessment

Overview

This document provides a quality assessment of Genome Analyzer results. The assessment is meant to complement, rather than replace, quality assessment available from the Genome Analyzer and its documentation. The narrative interpretation is based on experience of the package maintainer. It is applicable to results from the 'Genome Analyzer' hardware single-end module, configured to scan 300 tiles per lane. The 'control' results refered to below are from analysis of PhiX-174 sequence provided by Illumina.

Run Summary

Subsequent sections of the report use the following to identify figures and other information.

```
    Key
1   1
```

Read counts. Filtered and aligned read counts are reported relative to the total number of reads (clusters; if only filtered or aligned reads are available, total read count is reported). Consult Genome Analyzer documentation for official guidelines. From experience, very good runs of the Genome Analyzer 'control' lane result in 25-30 million reads, with up to 95% passing pre-defined filters.

ShortRead Quality Assessment

```
> # Check detail of selectedReads
> detail(selectedReads)
class:  ShortReadQ

sread:
DNAStringSet object of length 0

id:
BStringSet object of length 0
class: SFastqQuality
quality:
BStringSet object of length 0
>
>
>
> # Check reads of fqsample
> sread(fqsample)
DNAStringSet object of length 256:
      width seq
  [1]    36 GGACTTTGTAGGATACCCTCGCTTTCCTTCTCCTGT
  [2]    36 GATTTCTTACCTATTAGTGGTTGAACAGCATCGGAC
  [3]    36 GCGGTGGTCTATAGTGTTATTAATATCAATTTGGGT
  [4]    36 GTTACCATGATGTTATTTCTTCATTTGGAGGTAAAA
  [5]    36 GTATGTTTCTCCTGCTTATCACCTTCTTGAAGGCTT
  ...   ... ...
[252]    36 GTTTAGATATGAGTCACATTTTGTTCATGGTAGAGT
[253]    36 GTTTTACAGACACCTAAAGCTACATCGTCAACGTTA
[254]    36 GATGAACTAAGTCAACCTCAGCACTAACCTTGCGAG
[255]    36 GTTTGGTTCGCTTTGAGTCTTCTTCGGTTCCGACTA
[256]    36 GCAATCTGCCGACCACTCGCGATTCAATCATGACTT
>
> # Create myFil using polynFilter
> myFil <- polynFilter(threshold = 3, nuc = c("A"))
>
> # Apply your filter to fqsample
> filterCondition <- myFil(fqsample)
>
> # Use myFil with fqsample
> filteredSequences <- fqsample[filterCondition]
> filteredSequences
class: ShortReadQ
length: 13 reads; width: 36 cycles
```