Seminar III: R/Bioconductor GeneR

Amhed Missael Vargas Velazquez
avargas@lcg.unam.mx

September 25, 2009

What is GeneR?

GeneR is a package that allows direct use of nucleotide sequences within R software. Functions can be used to read and write sequences from main file formats (Embl, Genbank and Fasta) in order to perform a lot of manipulations and analyses.

What is GeneR?

GeneR is a package that allows direct use of nucleotide sequences within R software. Functions can be used to read and write sequences from main file formats (Embl, Genbank and Fasta) in order to perform a lot of manipulations and analyses.

- Authors
  L. Cottret, A. Lucas, E. Marrakchi, O. Rogier, V. Lefort, P. Durosay, A. Viari, C. Thermes; Y. d'Aubenton-Carafa.

What is GeneR?

GeneR is a package that allows direct use of nucleotide sequences within R software. Functions can be used to read and write sequences from main file formats (Embl, Genbank and Fasta) in order to perform a lot of manipulations and analyses.

- Authors
  L. Cottret, A. Lucas, E. Marrakchi, O. Rogier, V. Lefort, P. Durosay, A. Viari, C. Thermes; Y. d'Aubenton-Carafa.
- Maintainer
  Y. d'Aubenton-Carafa

What is GeneR?

GeneR is a package that allows direct use of nucleotide sequences within R software. Functions can be used to read and write sequences from main file formats (Embl, Genbank and Fasta) in order to perform a lot of manipulations and analyses.

- Authors
  L. Cottret, A. Lucas, E. Marrakchi, O. Rogier, V. Lefort, P. Durosay, A. Viari, C. Thermes; Y. d'Aubenton-Carafa.
- Maintainer
  Y. d'Aubenton-Carafa
- I think that Y. d'Aubenton-Carafa, entered the proyect at the end :)

GeneR is a very useful package which contains some functions for the manipulation of genetic data. It‘s similar to Biostrings[1], However, GeneR contains more functions and it used for different things. In addition, it is related to GeneRfold[2] package that allows the use of Vienna RNA library within R, meaning, tools for the prediction and comparison of RNA secondary structures.[3]
You can install the GeneR package in R using:

```
> source("http://bioconductor.org/biocLite.R")
> biocLite("GeneR")
```

[1] Biostrings was showed in the previous class by Isaac

[2] A package created by Y. d'Aubenton-Carafa, A. Lucas; C. Thermes, the same creator as the GeneR package XD

[3] It‘s an excellent package to talk about, and it is also interesting and easy to use.

What is it used for?

- Reading and writing sequences
  Fast sequence retrieving even from very large sequence databanks, in Fasta, Embl or Genbank formats.

What is it used for?

- Reading and writing sequences
  Fast sequence retrieving even from very large sequence databanks, in Fasta, Embl or Genbank formats.
- Handling sequences
  The usual copy-paste of parts of sequences or other manipulations can be performed by functions using vectors of strands and positions.

What is it used for?

- Reading and writing sequences
  Fast sequence retrieving even from very large sequence databanks, in Fasta, Embl or Genbank formats.
- Handling sequences
  The usual copy-paste of parts of sequences or other manipulations can be performed by functions using vectors of strands and positions.
- Analyzing sequences
  To count oligo-nucleotides by mono, di or tri, to look for exact word positions or to shuffle sequences.

What is it used for?

- Reading and writing sequences
  Fast sequence retrieving even from very large sequence databanks, in Fasta, Embl or Genbank formats.
- Handling sequences
  The usual copy-paste of parts of sequences or other manipulations can be performed by functions using vectors of strands and positions.
- Analyzing sequences
  To count oligo-nucleotides by mono, di or tri, to look for exact word positions or to shuffle sequences.
- Manipulation of regions on a chromosome
  Tools to easily compute any subregions (intergenic regions, exons or more sophisticated regions), without an exhaustive texture on a whole chromosome.

What is it used for?

- Performing bioinformatic jobs
  Functions related to genetic and structural aspects of the sequences : ORF localization, translation, or RNA secondary structure determination[4].

[4] with extention of GeneR: GeneRfold package

Working with sequences I

[5] I create a random sequence for the samples

```
> library(GeneR)
> seq <- randomSeq(prob = c(0.2, 0.3, 0.2, 0.3), letters =
+  c("T",  "C", "A", "G"), n = 30)
```

Insert a poly A into the end of the sequence

```
> seq <- insertSeq(seq, "AAAAAAAAAA", 30)
> seq
```

```
[1] "GAAACAGAGGCTCCTCTGGCTTCGTTTACAAAAAAAAAAAC"
```

[5] So sorry my friends, but this is a brief drescription of the GeneR, so im not going to explain each function. ; p

Compute the reverse complementary

```
> strComp(seq)
```

```
[1] "GTTTTTTTTTGTAAACGAAGCCAGAGGAGCCTCTGTTTC"
```

Count di-nucleotides[6]

```
> strCompoSeq(seq, wsize = 2)
```

```
      TT   TC   TA   TG TX   CT   CC  CA   CG CX AT   AC
[1,] 0.1 0.05 0.05 0.05  0 0.05 0.05 0.1 0.05  0  0 0.05 0.
       GG GX XT XC XA XG XX
[1,] 0.05  0  0  0  0  0  0
```

Translate the sequence string to a protein

```
> strTranslate(seq)
```

```
[1] "ETEAPLASFTKKK"
```

[6] It can be in groups from 1 to 15

Doing big jobs

Most of the functions in the GeneR package use buffers.

- Why use buffers

Doing big jobs

Most of the functions in the GeneR package use buffers.

- Why use buffers
- To work on large sequences (i.e. a whole chromosome).

Doing big jobs

Most of the functions in the GeneR package use buffers.

- Why use buffers
- To work on large sequences (i.e. a whole chromosome).
- In addition, you can buffer fasta sequences from Ncbi

Buffering the complete genome of Nanoarchaeum equitans[7] from Ncbi.

```
> seqNcbi("NC_005213", file = "toto.seq", submotif = TRUE
+ , type = "fasta")
```

```
[1] 1
```

```
> readFasta("toto.seq")
```

```
[1] 0
```

Size of the genome.

```
> sizeSeq()
```

```
[1] 490885
```

Looking for motifs[8].

```
> exactWord("ACTGA", seqno = 0, case.sensitive = TRUE)
```

```
[[1]]
 [1]    4925   6632   8764  12958  13693  18925  18940   1964
[11]   26758  31518  32702  33170  44284  44344  45825   4757
[21]   60992  69216  78148  97864 101865 107694 113767 12416
[31]  161255 165544 167140 167199 168805 172205 172462 17872
[41]  194550 201175 209660 216070 219809 227793 246409 24759
[51]  257148 262271 269888 273945 282269 294376 297681 30163
[61]  325389 330027 331853 332483 336450 355967 360722 36446
[71]  375564 384219 384256 384869 387519 389579 390623 39423
[81]  411202 411597 414553 419521 421865 422699 432651 44732
[91]  468659 478141 478817 490088 490136
```

[7] One of the most little genomes, i don‘t wanna break my computer

[8] Also, there is a function named getOrfs, that is supposed used to know where find Open Reading Frames, however, is not working :(

DNA TO RNA

```
> dnaToRna()
```

```
[1] 0
```

Or writing our new RNA file

```
> writeFasta(seqno = 0, file = "Nan_rna.fa", name =
+ "MyRNA", comment = "RNA generated by DNA
+ of Nanoarchaeum equitans",  append = TRUE)
```

```
[1] 1
```

You must remember, any function that uses the buffer, changes the content of the buffer.
We changed our DNA, so that if we use a getSeq you will see RNA

```
> getSeq(seqno = 0, from = 1, to = 30)
```

```
[1] "UCUCGCAGAGUUCUUUUUUGUAUUAACAAA"
```

You might prefer to change the number of the buffer for anything that you might do.

Bioinformatic Job

We already see in one of our class, how is constitute a bacterial genome...
So, why not use the functions to do a brief review the genome of the Rhizobium etli. We want to know:

Bioinformatic Job

We already see in one of our class, how is constitute a bacterial genome...
So, why not use the functions to do a brief review the genome of the Rhizobium etli. We want to know:

- The size

Bioinformatic Job

We already see in one of our class, how is constitute a bacterial genome...
So, why not use the functions to do a brief review the genome of the Rhizobium etli. We want to know:

- The size
- The GC content

Bioinformatic Job

We already see in one of our class, how is constitute a bacterial genome...
So, why not use the functions to do a brief review the genome of the Rhizobium etli. We want to know:

- The size
- The GC content
- A GC Skew of the genome

Buffering the sequence

```
> seqNcbi("NC_007761", file = "Retli.seq", submotif =
+  TRUE, type = "fasta")
```

```
[1] 1
```

```
> readFasta("Retli.seq")
```

```
[1] 0
```

The size

```
> sizeSeq()
```

```
[1] 4381608
```

The GC content

```
> GCcontent()
```

```
        pgc N
G 0.6127221 0
```

For the GC skew, i create a object with the size for sectionate the genome

```
> size <- sizeSeq()
```

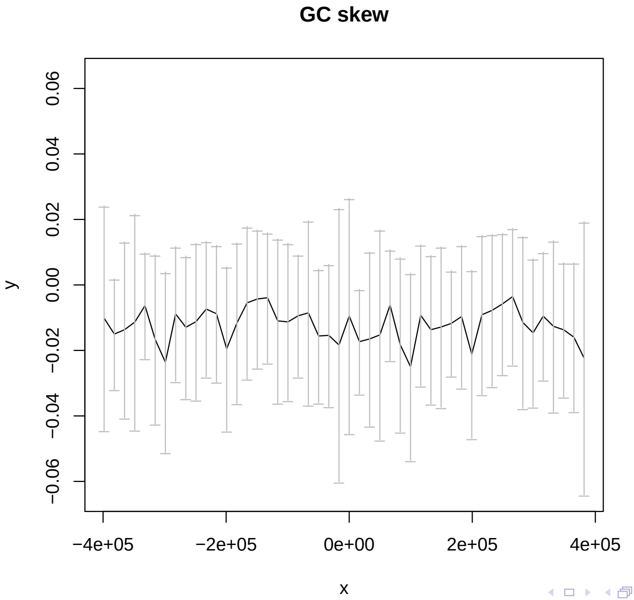
And now we use the function densityProfile

```
> dens <- densityProfile(ori = 398328 * (1:11), from = 1,
+  to = size,  seqno = 0, fun = seqSkew, nbinL = 24, nbinR
+  = 24, sizeBin = 16597)
```

At last, we plot :)

```
> plot(dens$skgc, main = "GC skew")

[1] 1
```


GC skew
y
0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06
−4e+05
−2e+05
0e+00
2e+05
4e+05
x

U - U

- GeneR has great tools:

U - U

- GeneR has great tools:
- To find a region in the genome

U - U

- GeneR has great tools:
- To find a region in the genome
- To manipulate sequences

U - U

- GeneR has great tools:
- To find a region in the genome
- To manipulate sequences
- To do large jobs

  As we see Gene R has the potential to be an excellent tool for conducting bioinformatics.

That’s All Folks

I encourage you to explore the Help Options of this package and to use them, they‘re user - friendly and fun XD .