



Introduction to `seqinR`

Pr. Jean R. LOBRY

August 3, 2017

Contents

1	About	2
1.1	ACNUC	2
1.2	 and CRAN	3
1.3	This document	3
1.4	<code>seqin</code> and <code>seqinR</code>	4
1.5	Getting started	4
1.6	Running  in batch mode	4
2	The learning curve	5
2.1	Wheel (the)	5
2.2	Hotline	5
2.3	Automation	5
2.4	Reproducibility	6
2.5	Fine tuning	6
2.6	Data as fast moving targets	8
2.7	<code>Sweave()</code> and <code>xtable()</code>	10
	References	13

1 About

1.1 ACNUC

ACNUC¹ was the first database of nucleic acids developed in the early 80's in the same lab (Lyon, France) that issued **seqinR**. ACNUC was published as a printed book in two volumes [5, 6] whose covers are reproduced in margin there. At about the same time, two other databases were created, one in the USA (GenBank, at Los Alamos and now managed by the NCBI²), and another one in Germany (created in Köln by K. Stüber). To avoid duplication of efforts at the european level, a single repository database was initiated in Germany yielding the EMBL³ database that moved from Köln to Heidelberg, and then to its current location at the EBI⁴ near Cambridge. The DDBJ⁵ started in 1986 at the NIG⁶ in Mishima. These three main repository DNA databases are now collaborating to maintain the INSD⁷ and are sharing data on a daily basis.

The sequences present in the ACNUC books [5, 6] were all the published nucleic acid sequences of about 150 or more continuous unambiguous nucleotides up to May or June 1981 from the journal given in table 1.

Journal name
<i>Biochimie</i>
<i>Biochemistry (ACS)</i>
<i>Cell</i>
<i>Comptes Rendus de l'Académie des Sciences, Paris</i>
<i>European Journal of Biochemistry</i>
<i>FEBS Letters</i>
<i>Gene</i>
<i>Journal of Bacteriology</i>
<i>Journal of Biological Chemistry</i>
<i>Journal of Molecular Biology</i>
<i>Molecular and General Genetics</i>
<i>Nature</i>
<i>Nucleic Acids Research</i>
<i>Proceedings of the National Academy of Sciences of the United States of America</i>
<i>Science</i>

Table 1: The list of journals that were manually scanned for nucleic sequences that were included in the ACNUC books [5, 6]

The total number of base pair was 526,506 in the two books. They were about 4.5 cm width. We can then compute of much place would it take to print the last GenBank release with the same format as the ACNUC book:

```
acnucbooksize <- 4.5 # cm
acnucbp <- 526506 # bp
```

¹A contraction of ACides NUCléiques, that is *NUcleic ACids* in french (<http://pbil.univ-lyon1.fr/databases/acnuc/acnuc.html>)

²National Center for Biotechnology Information

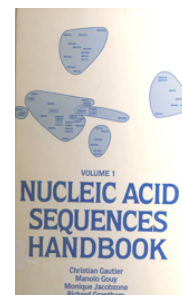
³European Molecular Biology Laboratory

⁴European Bioinformatic Institute

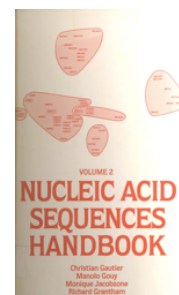
⁵DNA Data Bank of Japan

⁶National Institute of Genetics

⁷International Nucleotide Sequence Database (<http://www.insdc.org/>)



Cover of ACNUC book vol. 1



Cover of ACNUC book vol. 2



ACNUC books are about 4.5 cm width

```

choosebank("genbank") -> mybank
closebank()
mybank$details
[1] "          ****      ACNUC Data Base Content      ****          "
[2] "          GenBank Release 220 (15 June 2017) Last Updated: Jul 15, 2017"
[3] "236,647,372,946 bases; 202,357,489 sequences; 45,589,898 subseqs; 930,092 refers."
[4] "Software M. Gouy, Lab. Biometrie et Biologie Evolutive, Universite Lyon I "

unlist(strsplit(mybank$details[3], split=" ")) [1] -> bpbk
bpbk
[1] "236,647,372,946"

bpbk <- as.numeric(paste(unlist(strsplit(bpbk, split = ",")), collapse = ""))
widthcm <- acnucbooksize*bpbk/acnucbp
(widthkm <- widthcm/10^5)
[1] 20.22604

```

It would be about 20.2 kilometer long in ACNUC book format to print GenBank today (August 3, 2017). As a matter of comparison, our local university library buiding⁸ contains about 4 km of books and journals.



Our local library building in 2007 has a capacity of about 4 linear km of journals. That wouldn't be enough to store a printed version of GenBank. Picture by Lionel Clouzeau.

1.2 R and CRAN

R [8, 13] is a *libre* language and environment for statistical computing and graphics which provides a wide variety of statistical and graphical techniques: linear and nonlinear modelling, statistical tests, time series analysis, classification, clustering, etc. Please consult the R project homepage at <http://www.R-project.org/> for further information.

The Comprehensive R Archive Network, CRAN, is a network of servers around the world that store identical, up-to-date, versions of code and documentation for R. At compilation time of this document, there were 88 mirrors available from 48 countries. Please use the CRAN mirror nearest to you to minimize network load, they are listed at <http://cran.r-project.org/mirrors.html>, and can be directly selected with the function `chooseCRANmirror()`.

1.3 This document

In the terminology of the R project [8, 13], this document is a package *vignette*, which means that all code outputs present here were actually obtained by running them. The examples given thereafter were run under R version 3.4.1 (2017-06-30) on Thu Aug 3 21:58:01 2017 with Sweave [10]. There is a section at the end of each chapter called **Session Informations** that gives details about packages and package versions that were involved⁹. The last compiled version of this document is available at the `seqinR` home page at <http://seqinr.r-forge.r-project.org/>.

⁸Université de Lyon, F-69000, Lyon ; Université Lyon 1 ; Bibliothèque Universitaire Sciences, 18-25-27 Avenue Claude BERNARD, F-69622, Villeurbanne, France.

⁹Previous versions of R and packages are available on CRAN mirrors, for instance at <http://cran.univ-lyon1.fr/src/contrib/Archive>.

1.4 sequin and seqinR

Sequin is the well known software used to submit sequences to GenBank, **seqinR** [2] has definitively no connection with sequin. **seqinR** is just a shortcut, with no google hit, for "Sequences in R".

However, as a mnemotechnic tip, you may think about the **seqinR** package as the **R**eciprocal function of sequin: with sequin you can submit sequences to Genbank, with **seqinR** you can **R**etrieve sequences from Genbank (and many other sequence databases). This is a very good summary of a major functionality of the **seqinR** package: to provide an efficient access to sequence databases under R.

1.5 Getting started

You need a computer connected to the Internet. First, install **R** on your computer. There are distributions for Linux, Mac and Windows users on the CRAN (<http://cran.r-project.org>). Then, install the **seqinR** package. This can be done directly in an **R** console with for instance the command `install.packages("seqinR")`. Last, load the **seqinR** package with:

```
library(seqinR)
```

The command `lseqinR()` lists all what is defined in the package **seqinR**:

```
lseqinR()[1:9]
[1] "a"          "aaa"          "AAstat"       "acnucclclose"
[5] "acnucopen"  "al2bp"        "allistranks"  "alr"
[9] "amb"
```

We have printed here only the first 9 entries because they are too numerous. To get help on a specific function, say `aaa()`, just prefix its name with a question mark, as in `?aaa` and press enter.

1.6 Running **R** in batch mode

Although **R** is usually run in an interactive mode, some data pre-processing and analyses could be too long. You can run your **R** code in batch mode in a shell with a command that typically looks like :

```
unix$ R CMD BATCH input.R results.out &
```

where `input.R` is a text file with the **R** code you want to run and `results.out` a text file to store the outputs. Note that in batch mode, the graphical user interface is not active so that some graphical devices (*e.g.* `x11`, `jpeg`, `png`) are not available (see the R FAQ [7] for further details).

It's worth noting that **R** uses the XDR representation of binary objects in binary saved files, and these are portable across all **R** platforms. The `save()` and `load()` functions are very efficient (because of their binary nature) for saving and restoring any kind of **R** objects, in a platform independent way. To

give a striking real example, at a given time on a given platform, it was about 4 minutes long to import a numeric table with 70000 lines and 64 columns with the defaults settings of the `read.table()` function. Turning it into binary format, it was then about 8 *seconds* to restore it with the `load()` function. It is therefore advisable in the `input.R` batch file to save important data or results (with something like `save(mybigdata, file = "mybigdata.RData")`) so as to be able to restore them later efficiently in the interactive mode (with something like `load("mybigdata.RData")`).

2 The learning curve


Introduction

If you are used to work with a purely graphical user interface, you may feel frustrated in the beginning of the learning process because apparently simple things are not so easily obtained (*ce n'est que le premier pas qui coûte !*). In the long term, however, you are a winner for the following reasons.


2.1 Wheel (the)

Do not re-invent (there's a patent [9] on it anyway). At the compilation time of this document there were 11163 contributed packages available. Even if you don't want to be spoon-feed *à bouche ouverte*, it's not a bad idea to look around there just to check what's going on in your own application field. Specialists all around the world are there.


2.2 Hotline

There is a very reactive discussion list to help you, just make sure to read the posting guide there: <http://www.R-project.org/posting-guide.html> before posting. Because of the high traffic on this list, we strongly suggest to answer *yes* at the question *Would you like to receive list mail batched in a daily digest?* when subscribing at <https://stat.ethz.ch/mailman/listinfo/r-help>. Some *bons mots* from the list are archived in the  *fortunes* package.

2.3 Automation

Consider the 178 pages of figures in the additional data file 1 (<http://genomebiology.com/2002/3/10/research/0058/suppl/S1>) from [12]. They were produced in part automatically (with a proprietary software that is no more maintained) and manually, involving a lot of tedious and repetitive manipulations (such as italicising species names by hand in subtitles). In few words, a waste of time. The advantage of the  environment is that once you are happy with the outputs (including graphical outputs) of an analysis for species x, it's very easy to run the same analysis on n species.

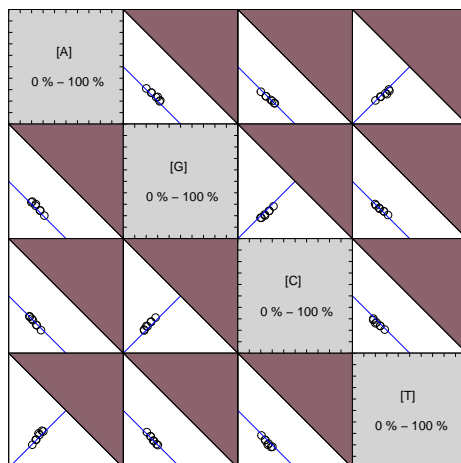
2.4 Reproducibility

If you do not consider the reproducibility of scientific results to be a serious problem in practice, then the paper by Jonathan Buckheit and David Donoho [1] is a must read. Molecular data are available in public databases, this is a necessary but not sufficient condition to allow for the reproducibility of results. Publishing the  source code that was used in your analyses is a simple way to greatly facilitate the reproduction of your results at the expense of no extra cost. At the expense of a little extra cost, you may consider to set up a RWeb server so that even the laziest reviewer may reproduce your results just by clicking on the "do it again" button in his web browser (*i.e.* without installing any software on his computer). For an example involving the `seqinR` package, follow this link <http://pbil.univ-lyon1.fr/members/lobry/repro/bioinfo04/> to reproduce on-line the results from [3].

2.5 Fine tuning

You have full control on everything, even the source code for all functions is available. The following graph was specifically designed to illustrate the first experimental evidence [14] that, on average, we have also $[A]=[T]$ and $[C]=[G]$ in single-stranded DNA. These data from Chargaff's lab give the base composition of the L (Ligth) strand for 7 bacterial chromosomes.

```
example(chargaff, ask = FALSE)
```



This is a very specialised graph. The filled areas correspond to non-allowed values because the sum of the four bases frequencies cannot exceed 100%. The white areas correspond to possible values (more exactly to the projection from \mathbb{R}^4 to the corresponding \mathbb{R}^2 planes of the region of allowed values). The lines correspond to the very small subset of allowed values for which we have in addition $[A]=[T]$ and $[C]=[G]$. Points represent observed values in the 7 bacterial

chromosomes. The whole graph is entirely defined by the code given in the example of the `chargaff` dataset (`?chargaff` to see it).

Another example of highly specialised graph is given by the function `tablecode()` to display a genetic code as in textbooks :

```
tablecode()
```

Genetic code 1 : standard							
TTT	Phe	TCT	Ser	TAT	Tyr	TGT	Cys
TTC	Phe	TCC	Ser	TAC	Tyr	TGC	Cys
TTA	Leu	TCA	Ser	TAA	Stp	TGA	Stp
TTG	Leu	TCG	Ser	TAG	Stp	TGG	Trp
CTT	Leu	CCT	Pro	CAT	His	CGT	Arg
CTC	Leu	CCC	Pro	CAC	His	CGC	Arg
CTA	Leu	CCA	Pro	CAA	Gln	CGA	Arg
CTG	Leu	CCG	Pro	CAG	Gln	CGG	Arg
ATT	Ile	ACT	Thr	AAT	Asn	AGT	Ser
ATC	Ile	ACC	Thr	AAC	Asn	AGC	Ser
ATA	Ile	ACA	Thr	AAA	Lys	AGA	Arg
ATG	Met	ACG	Thr	AAG	Lys	AGG	Arg
GTT	Val	GCT	Ala	GAT	Asp	GGT	Gly
GTG	Val	GCC	Ala	GAC	Asp	GGC	Gly
GTA	Val	GCA	Ala	GAA	Glu	GGA	Gly
GTG	Val	GCG	Ala	GAG	Glu	GGG	Gly

It's very convenient in practice to have a genetic code at hand, and moreover here, all genetic code variants are available :

```
tablecode(numcode = 2)
```

Genetic code 2 : vertebrate.mitochondrial							
TTT	Phe	TCT	Ser	TAT	Tyr	TGT	Cys
TTC	Phe	TCC	Ser	TAC	Tyr	TGC	Cys
TTA	Leu	TCA	Ser	TAA	Stp	TGA	Trp
TTG	Leu	TCG	Ser	TAG	Stp	TGG	Trp
CTT	Leu	CCT	Pro	CAT	His	CGT	Arg
CTC	Leu	CCC	Pro	CAC	His	CGC	Arg
CTA	Leu	CCA	Pro	CAA	Gln	CGA	Arg
CTG	Leu	CCG	Pro	CAG	Gln	CGG	Arg
ATT	Ile	ACT	Thr	AAT	Asn	AGT	Ser
ATC	Ile	ACC	Thr	AAC	Asn	AGC	Ser
ATA	Met	ACA	Thr	AAA	Lys	AGA	Stp
ATG	Met	ACG	Thr	AAG	Lys	AGG	Stp
GTT	Val	GCT	Ala	GAT	Asp	GGT	Gly
GTG	Val	GCC	Ala	GAC	Asp	GGC	Gly
GTA	Val	GCA	Ala	GAA	Glu	GGA	Gly
GTG	Val	GCG	Ala	GAG	Glu	GGG	Gly

TTT	Phe	TCT	Ser	TAT	Tyr	TGT	Cys
TTC	Phe	TCC	Ser	TAC	Tyr	TGC	Cys
TTA	Leu	TCA	Ser	TAA	Stp	TGA	Trp
TTG	Leu	TCG	Ser	TAG	Stp	TGG	Trp
CTT	Thr	CCT	Pro	CAT	His	CGT	Arg
CTC	Thr	CCC	Pro	CAC	His	CGC	Arg
CTA	Thr	CCA	Pro	CAA	Gln	CGA	Arg
CTG	Thr	CCG	Pro	CAG	Gln	CGG	Arg
ATT	Ile	ACT	Thr	AAT	Asn	AGT	Ser
ATC	Ile	ACC	Thr	AAC	Asn	AGC	Ser
ATA	Met	ACA	Thr	AAA	Lys	AGA	Arg
ATG	Met	ACG	Thr	AAG	Lys	AGG	Arg
GTT	Val	GCT	Ala	GAT	Asp	GGT	Gly
GTC	Val	GCC	Ala	GAC	Asp	GGC	Gly
GTA	Val	GCA	Ala	GAA	Glu	GGA	Gly
GTG	Val	GCG	Ala	GAG	Glu	GGG	Gly

Table 2: Genetic code number 3: yeast.mitochondrial.

As from `seqinR` 1.0-4, it is possible to export the table of a genetic code into a \LaTeX document, for instance table 2 and table 3 were automatically generated with the following `R` code:

```
tablecode(numcode = 3, latexfile = "../tables/code3.tex", size = "small")
tablecode(numcode = 4, latexfile = "../tables/code4.tex", size = "small")
```

The tables were then inserted in the \LaTeX file with:

```
\input{../tables/code3.tex}
\input{../tables/code4.tex}
```

2.6 Data as fast moving targets

In research area, data are not always stable. Consider figure 1 from [11] which is reproduced here in figure 1 page 11 here. Data have been updated since then, but we can re-use the same `R` code¹⁰ to update the figure:

```
data <- get.db.growth()
scale <- 1
lty Moore <- 1 # line type for Moore's law
date <- data$date
Nucleotides <- data$Nucleotides
Month <- data$Month
```

¹⁰This code was adapted from <http://pbil.univ-lyon1.fr/members/lobry/repro/lncs04/>.

TTT	Phe	TCT	Ser	TAT	Tyr	TGT	Cys
TTC	Phe	TCC	Ser	TAC	Tyr	TGC	Cys
TTA	Leu	TCA	Ser	TAA	Stp	TGA	Trp
TTG	Leu	TCG	Ser	TAG	Stp	TGG	Trp
CTT	Leu	CCT	Pro	CAT	His	CGT	Arg
CTC	Leu	CCC	Pro	CAC	His	CGC	Arg
CTA	Leu	CCA	Pro	CAA	Gln	CGA	Arg
CTG	Leu	CCG	Pro	CAG	Gln	CGG	Arg
ATT	Ile	ACT	Thr	AAT	Asn	AGT	Ser
ATC	Ile	ACC	Thr	AAC	Asn	AGC	Ser
ATA	Ile	ACA	Thr	AAA	Lys	AGA	Arg
ATG	Met	ACG	Thr	AAG	Lys	AGG	Arg
GTT	Val	GCT	Ala	GAT	Asp	GGT	Gly
GTC	Val	GCC	Ala	GAC	Asp	GGC	Gly
GTA	Val	GCA	Ala	GAA	Glu	GGA	Gly
GTG	Val	GCG	Ala	GAG	Glu	GGG	Gly

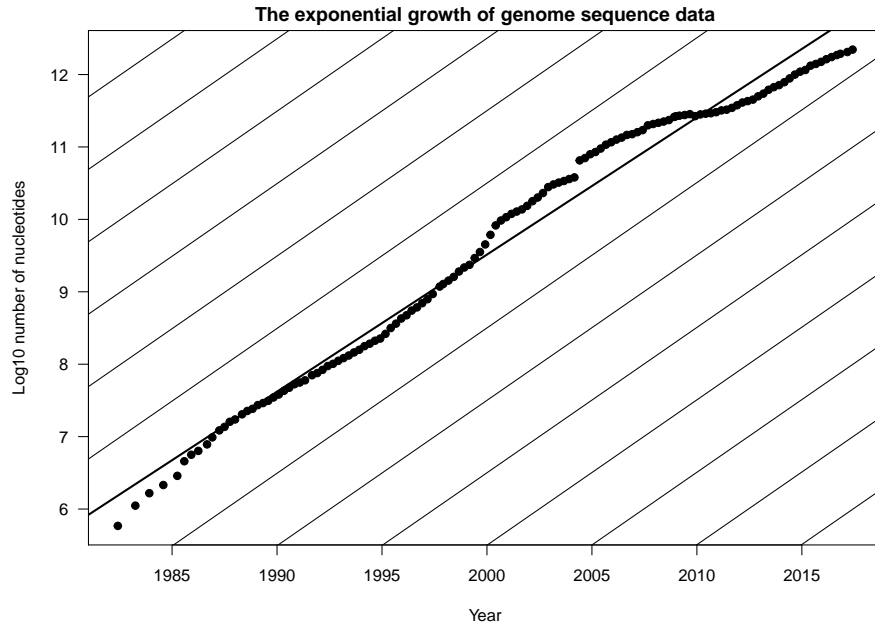
Table 3: Genetic code number 4: protozoan.mitochondrial+mycoplasma.

```

plot.default(date, log10(Nucleotides),
  main = "Update of Fig. 1 from Lobry (2004) LNCS, 3039:679:
  \nThe exponential growth of genome sequence data", xlab = "Year",
  ylab = "Log10 number of nucleotides", pch = 19, las = 1,
  cex = scale, cex.axis = scale, cex.lab = scale)
abline(lm(log10(Nucleotides) ~ date), lwd = 2)
lm1 <- lm(log(Nucleotides) ~ date)
mu <- lm1$coef[2]
dbt <- log(2)/mu
dbt <- 12 * dbt
x <- mean(date)
y <- mean(log10(Nucleotides))
a <- log10(2)/1.5
b <- y - a * x
lm10 <- lm(log10(Nucleotides) ~ date)
for (i in seq(-10, 10, by = 1)) if (i != 0)
  abline(coef = c(b + i, a), col = "black", lty = lty Moore)

```

Update of Fig. 1 from Lobry (2004) LNCS, 3039:679:



The doubling time is now 19.1 months.

2.7 Sweave() and xtable()

For \LaTeX users, it's worth mentioning the fantastic tool contributed by Friedrich Leish [10] called `Sweave()` that allows for the automatic insertion of \R outputs (including graphics) in a \LaTeX document. In the same spirit, there is a package called `xtable` [4] to coerce \R data into \LaTeX tables.

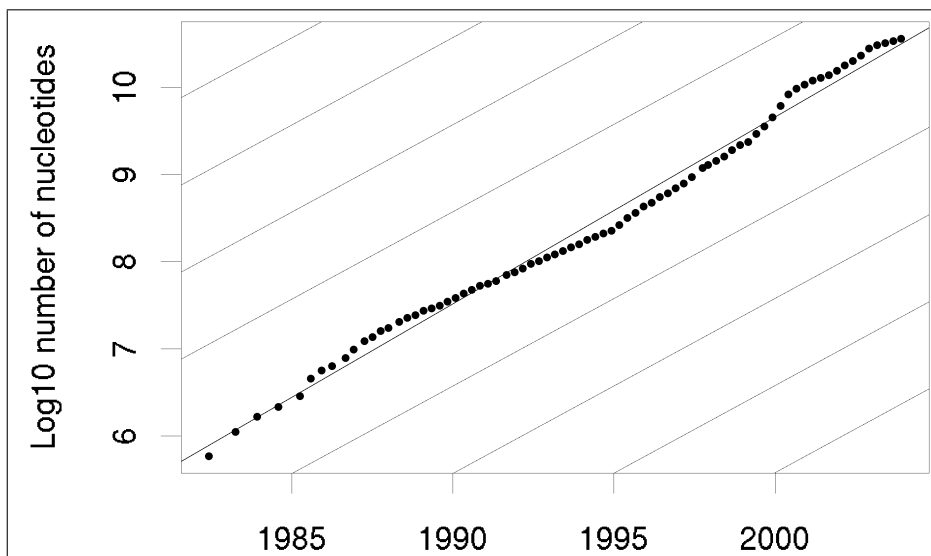




Figure 1: Screenshot of figure 1 from [11]. The exponential growth of genomic sequence data mimics MOORE's law. The source of data was the december 2003 release note (`realnote.txt`) from the EMBL database that was available at <http://www.ebi.ac.uk/>. External lines correspond to what would be expected with a doubling time of 18 months. The central line through points is the best least square fit, corresponding here to a doubling time of 16.9 months.

Session Informations

This part was compiled under the following  environment:

- R version 3.4.1 (2017-06-30), x86_64-apple-darwin15.6.0
- Locale: fr_FR.UTF-8/fr_FR.UTF-8/fr_FR.UTF-8/C/fr_FR.UTF-8/fr_FR.UTF-8
- Running under: macOS Sierra 10.12.5
- Matrix products: default
- BLAS: /Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRblas.0.dylib
- LAPACK: /Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRlapack.dylib
- Base packages: base, datasets, graphics, grDevices, grid, methods, stats, utils
- Other packages: ade4 1.7-6, ape 4.1, grImport 0.9-0, MASS 7.3-47, seqinr 3.4-5, tseries 0.10-41, XML 3.98-1.9, xtable 1.8-2
- Loaded via a namespace (and not attached): compiler 3.4.1, lattice 0.20-35, nlme 3.1-131, parallel 3.4.1, quadprog 1.5-5, quantmod 0.4-10, tools 3.4.1, TTR 0.23-1, xts 0.9-7, zoo 1.8-0

There were two compilation steps:

-  compilation time was: Thu Aug 3 21:58:14 2017
- \LaTeX compilation time was: August 3, 2017

References

- [1] J. Buckheit and D. L. Donoho. *Wavelets and Statistics*, chapter Wavelet and reproducible research. Springer-Verlag, Berlin, New York, 1995. A. Antoniadis editor.
- [2] D. Charif and J.R. Lobry. SeqinR 1.0-2: a contributed package to the R project for statistical computing devoted to biological sequences retrieval and analysis. In H.E. Roman U. Bastolla, M. Porto and M. Vendruscolo, editors, *Structural approaches to sequence evolution: Molecules, networks, populations*, Biological and Medical Physics, Biomedical Engineering, pages 207–232. Springer Verlag, New York, USA, 2007. ISBN 978-3-540-35305-8.
- [3] D. Charif, J. Thioulouse, J.R. Lobry, and G. Perrière. Online synonymous codon usage analyses with the ade4 and seqinR packages. *Bioinformatics*, 21(4):545–7, 2005.
- [4] D.B Dahl and *et al.* *xtable: Export tables to LaTeX or HTML*, 2005. R package version 1.3-0.
- [5] C. Gautier, M. Gouy, M. Jacobzone, and R. Grantham. *Nucleic acid sequences handbook. Vol. 1.* Praeger Publishers, London, UK, 1982. ISBN 0-275-90798-8.
- [6] C. Gautier, M. Gouy, M. Jacobzone, and R. Grantham. *Nucleic acid sequences handbook. Vol. 2.* Praeger Publishers, London, UK, 1982. ISBN 0-275-90799-6.
- [7] K. Hornik. *The R FAQ: Frequently Asked Questions on R (version 2.3.2006-07-13)*, 2006. ISBN 3-900051-08-9 <http://CRAN.R-project.org/doc/FAQ/>.
- [8] R. Ihaka and R. Gentleman. R: A language for data analysis and graphics. *J. Comp. Graph. Stat.*, 3:299–314, 1996.
- [9] J. Keogh. Circular transportation facilitation device, 2001.
- [10] F. Leisch. Sweave: Dynamic generation of statistical reports using literate data analysis. *Proceedings in Computational Statistics*, Compstat 2002:575–580, 2002.
- [11] J.R. Lobry. Life history traits and genome structure: aerobiosis and G+C content in bacteria. *Lecture Notes in Computer Sciences*, 3039:679–686, 2004.
- [12] J.R. Lobry and N. Sueoka. Asymmetric directional mutation pressures in bacteria. *Genome Biology*, 3(10):research0058.1–research0058.14, 2002.
- [13] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2016.

- [14] R. Rudner, J.D. Karkas, and E. Chargaff. Separation of microbial deoxyribonucleic acids into complementary strands. *Proceedings of the National Academy of Sciences of the United States of America*, 63:152–159, 1969.