SeqinR release notes

February 24, 2018

Introduction

These release notes are listed in reverse chronological order: most recent on top.

3.4 series

release 3.4-6

- As suggested by e-mail on 22-FEB-2018 by Haruo Suzuki the `read.fasta()` function has gained a new argument `whole.header` to control whether the header line should be truncated or not to produce the sequence name. The function `read.alignment()` now transmits extra arguments to `read.fasta()` in the fasta format case. An example is given the `read.alignment()` documentation.

release 3.4-5 (current on CRAN)

Minor CRAN submission release which supersedes 3.4-4 that failed to build properly on R-forge for an unknown reason.

release 3.4-4

Minor CRAN submission release to add a period at the end of the Description field in the `DESCRIPTION` file as requested by e-mail by Uwe Ligges on 2017-08-01 00:04.

release 3.4-3

- Minor CRAN submission release to add a sentence in the `DESCRIPTION` file as requested by e-mail by Uwe Ligges on 2017-07-31 08:12.
- Adding a `NEWS` file linking to this[1] file.

[1] `http://seqinr.r-forge.r-project.org/src/appendix/releasenotes.pdf`

release 3.4-2

Minor CRAN submission release to add a DOI to ACNUC [9] in the `DESCRIPTION` file as requested by e-mail by Uwe Ligges on 2017-07-30 21:17.

release 3.4-1

Minor CRAN submission release to add a reference to ACNUC [9] in the `DESCRIPTION` file as requested by e-mail by Uwe Ligges on 2017-07-25 23:19.

release 3.4-0

- New test file `kaks-torture.fasta` and corresponding dataset `kaksTorture` to check results of function `kaks()`.
- Function `read.alignment()` can now handle legacy fasta format with commented lines starting with a semicolon.
- Function `kaks()` has gained a new argument `rmgap` to control gap removal option. The C code was modified to increase numeric stability.
- As pointed by e-mail on 27-JUN-2017 by Sylvain Charlat the function `kaks()` could return non-finite values especially with short sequences. This is no more the case and a routine test checks now that computing the K_a and K_s values between all possible pairs of codons doesn't yield non-finite values.
- The routine check in the documentation of the `kaks()` function with `data(AnoukResult)` is active again.

3.3 series

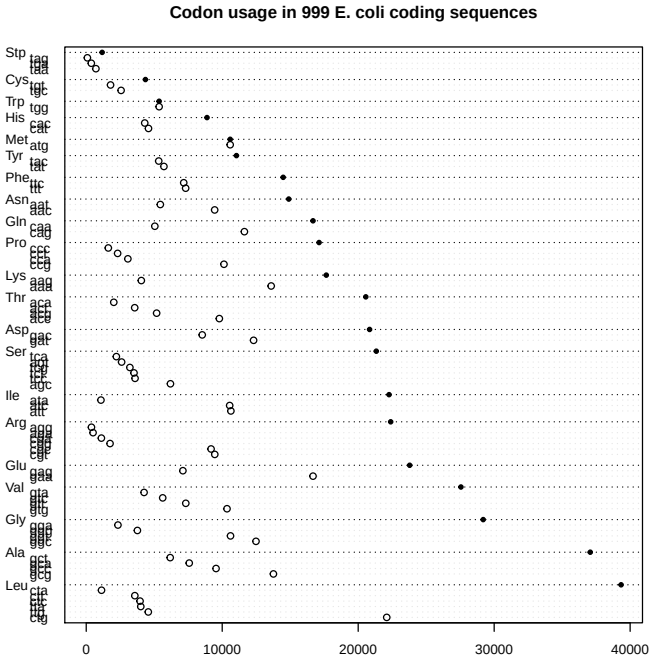
release 3.3-6 (06-APR-2016)

- Outdated URL in the documentation of function `oriloc()` were fixed.
- Addition of `packagename_init.c` and modifications for registered routines.

release 3.3-4

- As pointed by e-mail on 14-OCT-2016 by Christine Oger there was a bug in the function `dotchart.uco()` yielding points with an excessive size. This now fixed, for instance:

```
data(ec999)
ec999.uco <- rowSums(sapply(ec999, uco, index="eff"))
dotchart.uco(ec999.uco, main = "Codon usage in 999 E. coli coding sequences")
```



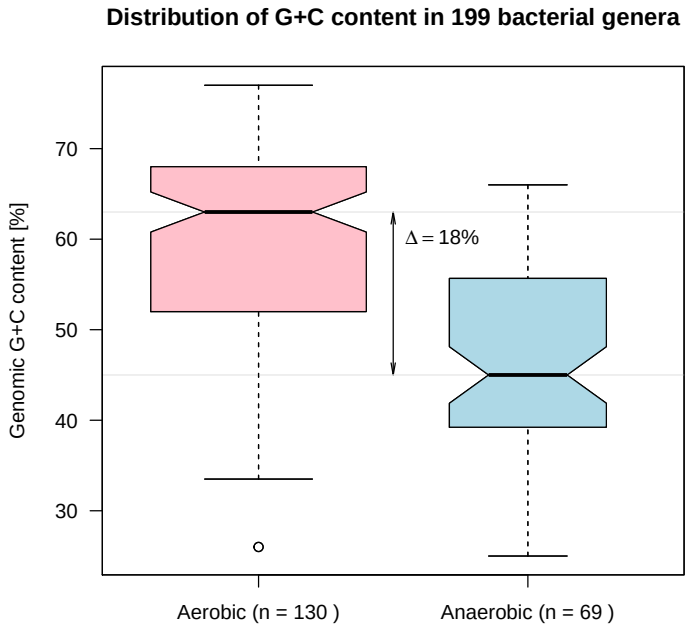
release 3.3-3 (13-OCT-2016)

- As requested by Kurt Hornik on 12-OCT-2016 the name `seqinR` was changed to `seqinr` in the `CITATION` file.

- New dataset `gc02` that was used in Naya *et al* [22]. For instance to show the dramatic effect of aerobiosis on genomic G+C content in bacteria:

```
data(gc02)
vby <- function(...) as.vector(by(...))
first <- function(x) as.character(x[1])
GCbyGenus <- with(gc02, data.frame(
  Genus = vby(Genus, Genus, first),
  GC = vby(GC, Genus, mean),
  aerobiosis = vby(aerobiosis, Genus, first),
  n.species = vby(GC, Genus, length)))
with(GCbyGenus, {
  mybxp <- boxplot(GC~aerobiosis, xaxt = "n", yaxt = "n", ann = FALSE,
    names = c(paste("Aerobic (n =", sum(aerobiosis == "Aerobic"), ")"),
    paste("Anaerobic (n =", sum(aerobiosis == "Anaerobic"), ")")),
    varwidth = TRUE, notch = TRUE)
  y1 <- median(GC[aerobiosis == "Anaerobic"])
  y2 <- median(GC[aerobiosis == "Aerobic"])
  arrows(1.5, y1, 1.5, y2, code = 3, angle = 10, length = 0.1)
  abline(h = y1, col = grey(0.9))
  abline(h = y2, col = grey(0.9))
  text(1.5, 60, bquote(paste(Delta == .(y2 - y1), "%")), pos = 4)
  bxp(mybxp, varwidth = TRUE, notch = TRUE, add = TRUE,
```

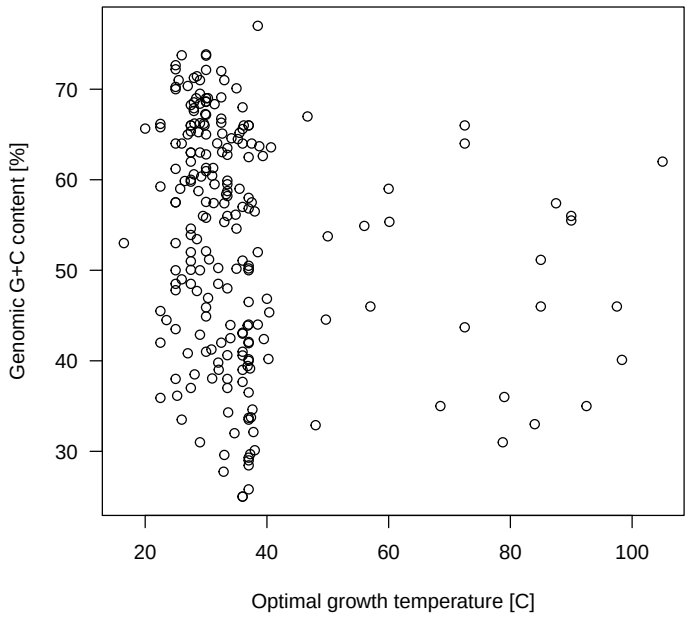
```
    main = paste("Distribution of G+C content in", length(GC), "bacterial genera"),
    las = 1, ylab = "Genomic G+C content [%]", boxfill = c("pink", "lightblue"))}
)
```



- New dataset `gcT` that was used in [5]. For instance to reproduce figure 2:

```
data(gcT)
with(gcT[["genus"]], plot(Topt, GC, las = 1,
  main ="Figure 2 from Galtier & Lobry 1997 (n = 224 genera)",
  xlab = "Optimal growth temperature [C]", ylab = "Genomic G+C content [%]"))
```

Figure 2 from Galtier & Lobry 1997 (n = 224 genera)



release 3.3-2 (07-OCT-2016)

- As pointed by e-mail on 04-OCT-2016 by Paulo Jorge Moura Pinto da Costa DIAS three new genetic code were missing. They are now included.

release 3.3-1 (07-SEP-2016)

release 3.3-0 *in memoriam* 2016-07-14

- Function `extractseqs()` has gained an extra argument `zlib` defaulting to `FALSE` so that it can be used on any platform. This follows from a request by Sam BORSTEIN on 10-JUL-2016 on how to extract D-loop/control region in mitochondrial genomes. An example is now given in the FAQ.

- Function `read.alignment()` has been modified according to Matthew KRAUSE in order to read abreviated path like '~login/alignment.txt' as filename. The C source code as been modified as well according to Matthew KRAUSE in order to avoid problems due to corrupted or empty files.

3.2 series

release 3.2-0 (07-JUL-2016)

- Images have been inserted in the documentation for datasets `aacost`, `chargaff`, `m16j`, `waterabs` and for the function `dia.db.growth()`.
- Function `gb2fasta()` now uses a local file example.
- As pointed out by e-mail on 11-JUN-2016 by Haruo Suzuki a call to `AAstat()` function may yield uninformative warning messages. The culprit is the `computePI()` function. This is now fixed.
- As from **seqinR** 3.2-0 we are switching to a lazy sticky scheme for **seqinR** release numbers. Release 3.2-x means that **seqinR** was checked against R release 3.2-y. In short, we are trying to follow R major revision numbers. This does not mean that you need a brand new version of R to run **seqinR**, for instance at this point (`2016-06-17`) you need at least R version 2.10.0 that was released in 2009. This new numbering scheme is just a matter of convenience.

3.1 series

release 3.1-5 (08-JUN-2016)

- As pointed out by e-mail on 30-MAY-2016 by Haruo Suzuki a call to `getLength(ec999)` yielded spurious output and many warnings. This is now fixed.
- As pointed out by e-mail on 30-MAY-2016 by Haruo Suzuki there was a bug in the documentation of the functions `recstat()`, `draw.recstat()`, `test.co.recstat()` and `test.li.recstat()`. They were all looking for data in a package `seqinr2` that doesn't exist. This is now fixed and the `dontrun` directive has been removed to detect automatically any further problem.
- As pointed out by e-mail on 25-MAY-2016 by Haruo Suzuki the `read.fasta()` function can import sequences directly from *local* gzipped files. A new `smallAA.fasta.gz` file has been added to document this in the examples of the `read.fasta()` function. This is however no more true if you try to read directly the sequences from a compressed file accessed via its URL. A workaround now given in the manual is to use a construct like `read.fasta(gzcon(url(myurl)))`.
- As pointed out by e-mail on 12-MAY-2016 by Haruo Suzuki the documentation for the `rho()` function was misleading because in the refered article [11] the statistic was computed from the sequence concatenated with its inverted complement. This is now fixed.

release 3.1-5 (08-JUN-2016)

Undocumented for this release.

release 3.1-4

- As pointed out by e-mail on 27-APRIL-2016 by Matthias DÖRING, R and Y were not correctly implemented in the `comp()` function. This is now fixed.
- As suggested by e-mail on 23-DECEMBER-2014 by Matt HUSKA, ade4 package has been switched from "Depends" to "Imports" in the `DESCRIPTION` file.

release 3.1-1

- Removal of zlib code and headers with help from Prof. RIPLEY.

release 3.1-0

- As suggested by e-mail on 19-NOVEMBER-2014 by Tang Chin CHEUNG, examples involving the `query()` function have been modified according to the new features of R-3.1.2. Indeed since R-3.1.2 it is not possible to change an object belonging to the global environment from a package, as spotted by e-mail on 4-NOVEMBER-2014 by T.J. AGIN.

3.0 series

release 3.0-11

- In `query()`, `NS=taxon_name` and `NK=keyword_name` are now documented. The manual was also updated.
- The broken default link in `get.db.growth()` has been fixed so that now `dia.db.growth()` works as well.
- Function `write.fasta()` has gained an `as.string` argument so that it can handle sequences provided as strings instead of vectors of single character.

release 3.0-10

- As pointed out by e-mail on 5-MAY-2014 by Jan-Hendrik TROESEMEIER, the `read.alignment()` function was sending segfault with non-mase data. This is now fixed in the src/alignment.c source according to Jan-Hendrik TROESEMEIER suggestion.

release 3.0-9

- As suggested by e-mail on 17-JANUARY-2014 by Peter Hraber, the `dist.alignment()` function include a new option for nucleotide sequences: if set to 1, gaps will be counted in the identity measure. (in case there is a gap aligned with and a non-gap, the number of difference is incremented.).

release 3.0-7

- As pointed out by e-mail on 04-OCTOBER-2013 by Jeremy Shearman, the `comp()` function was misleading when N's (instead of n's) were present in the sequence. This is now fixed.
- As pointed out by e-mail on 29-MAY-2013 by Domenico Cozzetto, the `read.alignment()` function was chopping sequence names after the first space . This is now fixed for FASTA, CLUSTAL and MASE formats.

release 3.0-6

- As pointed out by e-mail on 13-AUGUST-2012 by Grabor Grothendieck, the pipe caracter was not correctly processed in `stresc()` function. This is now fixed.
- As suggested by e-mail on 8-FEBRUARY-2012 by Juanjo Abellan, the `kaks()` function has been modified : new verbose option for display values of L0,L2,L4,A0,A2,A4,B0,B2,B4 has been added.

release 3.0-5 (13-OCT-2011)

- As pointed out by e-mail on 3-FEBRUARY-2012 by Dave Gerrard, the `consensus()` function returns NA for all invariant sites when threshold is 1. Changing 'superior' into 'superior or equal' in the consenus function fixed this bug.

release 3.0-4 (24-SEP-2011)

Was published on CRAN but no release note are available.

release 3.0-3 (07-APR-2011)

Was published on CRAN but no release note are available.

release 3.0-1 (15-NOV-2010)

Was published on CRAN but no release note are available.

release 3.0-0 (30-AUG-2010)

- As pointed out by Leonor Palmeira on the `rpourlesnuls` diffusion list on 20-MAY-2010 there was no constructor for objects of class alignment. There is now a `as.alignment()` function.

2.0 series

release 2.0-9 (20-MAR-2010)

- As pointed out by Avril Coghlan on the seqinR diffusion list on 17-MAR-2010 there was a bug in the `getAnnot()` function. This is now fixed.
- As suggested by Avril Coghlan on the seqinR diffusion list on 02-MAR-2010 the function `rho()` has gained a `wordsize` argument.
- The argument `word` in function `count()` is now more explicitely called `wordsize`.
- The example section in file `read.alignment.Rd` has gained a new quality control sanity check.
- The `File` argument that was deprecated since seqinR release 1.1-3 in function `read.alignment()` is no more valid. Just use `file` instead.
- As pointed by Darren Obbard on the seqinr diffusion list on 05-MAR-2010 there was a memory leak problem when calling the `read.alignment()` function with the fasta format. This is now fixed for the fasta format, but the remaining formats have not been checked for this problem.

release 2.0-8 (28-FEB-2010)

- As pointed by Oliver Clay and Lionel Guy on the seqinr diffusion list on 19-FEB-2010 there was a bug in `getSequence.list()` function that confused `write.fasta()` when all sequences were of the same length (a similar bug was reported by Yann Lesecque on 30-MAR-2009 for the `getTrans()` function). This is now fixed.
- The message printed when function `where.is.this.acc()` fails to find a database with a given accession number for a sequence is now completed to warn the user that (s)he may have supplied a sequence name instead of a genuine accession number.
- The title in the documentation for the function `write.fasta()` was changed to make clear that more than one sequence can be written at once. The function now does not return anything instead of `NULL` previously. The argument `file.out` was moved to the left so that it is easier now to use it by position during function call.

release 2.0-7 (17-NOV-2009)

- A new utility function `where.is.this.acc()` was introduced to loop over all availabale ACNUC databases to look for a given sequence accession number. This is useful when you have a sequence accession number and you don't know in which database it is present. The documentation of the function `choosebank()` was also changed to make a link to this function. As suggested by Avril Coghlan, the function has an argument `stopAtFirst` defaulting to `TRUE` that stops the search at the first database found with the given accession number.

- As pointed out 05 Nov 2009 by Darren Obbard on the seqinr diffusion list the argument `forceToLower = FALSE` in function `comp()` was not honored. This is now fixed and a new sanity check was added in the example section of the documentation of the function.

- Documentation for the function `uco()` for codon usage table computation was updated with new bibliographical references [20, 33].

- As basic regular expressions are defunct since R 2.11, the `extended` argument in functions `words.pos()` and `trimSpace()` was no more necessary. It is now deleted.

release 2.0-6 (18-OCT-2009)

- The old argument `File` in function `read.fasta()` that was deprecated since release 1.1-3 is no more valid. Just use `file` instead.

- New function `stutterabif()` to estimate stutter ratio.

- Function `plotabif()` has a new default value for its `ylim` argument: `c(min(y), max(y))` now instead of `c(0, max(y))` previously to help ploting data with a highly negative baseline.

- Function `peakabif()` now returns in addition an estimate of the baseline value.

- New utility `baselineabif()` to estimate the baseline value.

- There was time shift of one datapoint unit for the peak locations returned by the `peakabif()` function, this is now fixed and the documentation is more explicit for the units used.

- New utility function `fastacc()` to compute the number of alleles in common between a genetic profile and a database of genetic profiles.

release 2.0-5 (18-SEP-2009)

- New utility function `circle()` to draw a circle.
- Two more examples of files to be imported with the `readBins()` and `readPanels()` functions are now available in the `abif` folder: `NGM_Bins.txt` and `NGM_Pa.txt`, respectively.
- New function `plotPanels()` to plot amplicon size ranges of STR kits data.
- New utility function `col2alpha()` to add a transparency chanel to a standard R color.
- New ABIF example file `samplefsa2ps.fsa` used in the `read.abif()` function to reproduce figure 1A from [14].
- New function `move()` aliased as `mv()` to rename an object without deep copy.
- New function `swap()` to exchange two objects.

release 2.0-4 (01-JUL-2009)

- Configuration files to be imported by the `readBins()` function may have trailling tabulations, as for instance in the test file `Prototype_PowerPlex_EP01_Bins.txt` for allele 9 at locus `D3S1358` and for allele 14 at locus `D12S391`. This was a source of trouble during importation. This is now fixed and the above mentioned file is used as a quality control. A warning is now issued if the number of columns in the data.frame corresponding to a locus is not 4 as expected.
- Configuration files to be imported by the `readPanels()` function may have more than one tabulation separator between two data items in a way that could be different from one line to another one. There is an example of such a case in the test file `Prototype_PowerPlex_EP01_Pa.txt` where locus `D10S1248` and `D22S1045` are followed by a single tabulation when all remaining loci are followed by two tabulations. This was a source of trouble during importation. This is now fixed by preprocessing the input so that all consecutive tabulations are replaced by a single one. The above mentioned test file is now used as a quality control.

release 2.0-3 (28-APR-2009)

- As pointed out on the seqinr diffusion list on 23-APR-2009 by Darren Obbard there was an obscure error message when function `kaks()` was called with an alignment such that the number of nucleotides was not a multiple of 3 after gap removal. This check was partial as an alignment with out-of-frame gaps but with a total number of gaps multiple of 3 was not detected. The new behaviour is that if at least one non ACGT base

is found in a codon, then the whole codon is forced to a gap codon (`--`). The documentation of the function has been clarified accordingly, and a new alignment file `DarrenObbard.fasta` added in the `sequences` folder to check this new behaviour.

- Function `readBins()` is now more tolerant when there is an extra column with possibly empty fields in data by forcing the `fill` argument of `read.table()` function to `TRUE`.

- As pointed out by e-mail on 30-MAR-2009 by Yann LESECQUE there was a bug in the `getTrans()` function: when applied to a list of sequences with all the same length the returned result was a matrix instead of a list. This is now fixed.

- New utility functions `readPanels()` and `readBins()` to import data from GeneMapper configuration files. Four example files are now in the `abif` folder.

- Function `peakabif()` now returns the heights and surfaces of peaks in addition to their location.

- New utility function `al2bp()` to convert a forensic microsatellite allele name into its length in base pairs. Conventions used to name forensic microsatellite alleles (STR) are described in Bar *et al.* (1994) [1]. The name `9.3` means for instance that there are 9 repetitions of the complete base oligomer and an incomplete repeat with 3 bp.

release 2.0-2 (06-FEV-2009)

- New ABIF format related functions: `plotabif()` to plot electrophoregrams with optonial internal size standard and optional allelic ladder, `peakabif()` to locate peaks in electrophoregrams, `plotladder()` to display an observed allelic ladder.

- New datasets `gs500liz` for size standards, `identifiler` for allelic ladder names, `ECH` for allelic ladder raw data and `JLO` for forensic genetic profile raw data. The last one is now used as a quality check for the `read.abif()` function.

- A new folder called `abif` has been created under the `inst` folder. The purpose of this folder is to contain examples of files in ABIF format so that the results of the `read.abif()` function can be checked against expected results for quality check. It contains for now two duplicated genetic profiles and two allelic ladders from the same batch experiment.

release 2.0-1 (14-DEC-2008)

- The useless `itemize` in the argument section of documentation file `stresc.Rd` is now deleted.

- In function `words.pos()` the default value for parameter `extended` was changed from FALSE to TRUE to avoid warnings.
- New experimental function `read.abif()` to import files in ABIF format (`*.fsa`, (`*.ab1`).

release 2.0-0 (27-OCT-2008)

- New draft chapter about making RISA *in silico* added.
- Objects from class `qaw` created after a call to the `query()` function have gained a new generic `print` method to focus on the most important information: number of sequences in the list, list type and the corresponding request.
- Function `query()` now allows a missing `listname` argument. In this case, `list1` is used to store the result.
- Function `autosocket()` has been changed to behave more friendly with outdated R versions. This is essentially a backward compatibility issue that will not be maintained in the future. The function `autosocket()` works hard to check that everything is OK with the last opened database, especially with the socket infos available in `banknameSocket$socket` thru its `summary()` generic. In old R versions (*e.g.* 2.6.2) this was returning `socket` instead of `sockconn` for the class, yielding an error in seqinR 1.1-7. The old result is now allowed but a warning is issued.

The 2.0 series started in summer 2008 along with the moving of the seqinr sources on R-forge.

1.1 series

release 1.1-7 (29-JUL-2009)

- As suggested by Kurt Hornik two extra `cr` in the documentation file for `ec999` were deleted.
- Function `read.fasta()` has gained four new arguments (*viz.* `bfa`, `sizeof.longlong`, `endian`, `apply.mask`) to read DNA binary fasta files in MAQ format. There is a new `ct.bfa` file in the `sequences` folder to check for the MAQ format reading.
- New dataset `pK` for the values for the side chain of charged amino acids from various sources compiled by Joanna Kiraga [13].
- Function `words.pos()` has gained new arguments that are passed to `regexpr()` including the dot-dot-dot argument in case of need in the future. The documentation has been modified to better explain the difference with the standard `gregexpr()` function.

- As pointed by e-mail on 28 May 2008 by Kim MILFERSTEDT a function to compute the consensus for a set of aligned sequences would be helpful. There is now a function `consensus()` aliased to `con()` for this. The input is either an object from class `alignment` or a matrix of characters. The output is either a consensus sequence (using the majority rule, the majority rule with a threshold, or IUPAC symbols for RNA and DNA sequences) or a profile, that is a matrix with the count of each possible character at each position in the alignment.

- In the documentation of the `read.alignment()` function a link was added to the `read.nexus()` function from the `ComPairWise` package [28].

- New function `bma()` to find the IUPAC symbol corresponding to a nucleic sequence.

- New function `as.matrix.alignment()` to convert an alignment into a object of class `matrix`.

- The encoding of line ends in the example file `test.mase` is now an unix-like one.

- As pointed by e-mail on 31 May 2008 by Marie SÉMON there was no convenient function to compute the Codon Adaptation Index [31]. A new function `cai()` was introduced with the aim of reproducing exactly the results from the program `codonW` that was written by John PEDEN during his PhD thesis [26] under the supervision of P.M. SHARP (the most authorative source for CAI computation). A new dataset `caitab` that was hard-encoded in `codonW` for the `w` values for some species (*viz Escherichia coli*, *Bacillus subtilis*, *Saccharomyces cerevisiae*) was added. Care was taken to credit original sources. The *E. coli* data that was uncredited is from [31]. The *B. subtilis* data that was uncredited is from [32] (see the note of caution in `?caitab` before using this one directly to compute CAI in *B. subtilis*). The *S. cerevisiae* data that was credited to [30] dates back from [31]. A new text file `scuco.txt` produced by `codonW` was added in the `sequences` folder to check that the CAI results from `cai()` are consistents with thoses from `codonW` version 1.4.4 (03-MAR-2005). This legacy file is used in the example section of the `cai()` function.

release 1.1-6 (21-MAY-2008)

- The construct `get(getOption("device"))(width = 18, height = 11)` that was used in the example section for `data(prochlo)` is no more valid since R 2.8.0 (fall 2008). The example has been restricted to work only with `X11`, `windows` and `quartz` devices.

- As pointed by e-mail on 12 May 2008 by Indranuj MUKHERJEE there was a bug in the function `oriloc()`: when called with a `gbk = NULL` argument the function was trying to remove non-existent files, yielding an error. The

bug has been fixed and the documentation of the function `oriloc()` has been extended to better explain how to use the arguments `seq.fasta` and `gbk`.

- A reference to [8] was missing in the documentation of function `zscore()` for the codon model.

- As suggested by e-mail on 11 Mar 2008 by Christian Gautier, the function `count()` has gained a new argument `by` to control the window step, allowing for instant to count dinucleotides in codon position III-I in a coding sequence. The example section of the function documentation has been extended to give an example of counting dinucleotides in position III-I.

```
alldinuclIIIpI <- s2c("NNaaNatNttNtgNgtNtcNctNtaNagNggNgcNcgNgaNacNccNcaNN")
(resIIIpI <- count(alldinuclIIIpI, word = 2, start = 2, by = 3))
```

```
aa ac ag at ca cc cg ct ga gc gg gt ta tc tg tt
 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
```

```
stopifnot(all( resIIIpI == 1))
```

- Function `reverse.align()` has gained two arguments `forceDNAtolower = TRUE` and `forceAAtolower = FALSE` that are passed to the functions used to read the sequences. There is now a new dataset `revaligntest` used to check the result in the example section of `reverse.align()`.

- As pointed by e-mail on 21 Feb 2008 by Oliver Keatinge Clay function `modifylist()` failed to scan in GenBank FEATURES annotation lines. There is now a new function called `prepgetannots()`, aliased to `pga()`, that allows to set up the annotation lines to be scanned. Called with default arguments, this function turns on all annotation lines for scan. This function can also be used to set up partly the annotation lines to be returned by `getAnnot()`.

- Function `choosebank()` has gained four arguments (`server`, `blocking`, `open`, `encoding`) that are passed to `socketConnection()`. The value of the argument `verbose` is now passed to `clientid()` which knows now how to handle it. The `encoding` argument was introduced to fix a localization bug on Mac OS X which symptom was a cryptic error message in `if (res[1] != "0") {` after a call to `choosebank()`. The culprit was an `option(encoding = "latin1")` that was set up before the call to `choosebank()` who called `socketConnection()` with its default `encoding = getOption("encoding")`, preventing `readLines()` to read from the socket. The bug was fixed by opening the socket with the native encoding, which is the current default.

- As pointed by e-mail on 15 Jan 2008 by Stefanie Hartmann, the argument `frame` in function `count()` was misleading for someone with a molecular biology background. The argument has been replaced by `start`. The

old argument name is maintained as an alias for backward compatibility. The example section has been extended to give an example with the complete human mitochondrion sequence, the corresponding fasta file (`humanMito.fasta`) has been added in the `sequences` directory.

release 1.1-5 (18-DEC-2007)

Minor release to fix mainly problems in the documentation.

- The argument section was empty in `autosocket.Rd`.
- The details section was empty in `countfreelists.Rd` and `draw.oriloc.Rd`.
- The value section was empty in `gbk2g2.Rd`. The corresponding function was changed to use a local file for the demo.
- The description section was missing in `getFrag.Rd`, `getLength.Rd`, `getName.Rd`, `getSequence.Rd`.
- Documentation of the function `dia.bactgensize()` to plot the distribution of bacterial genome size from GOLD data has been ammended to credit sources [15, 2, 17, 16]. It has gained a new argument `maxgensize` defaulting to 20000 to remove outliers. It has also gained a new argument `source` for the file to look for raw data, defaulting to an (outdated) local copy so that the function can be called even when there is no internet connection.

release 1.1-4 (10-Dec-2007)

Minor release to fix problems found by Kurt Hornik.

- In the `DESCRIPTION` file `License: GPL (>= 2)` instead of `License: GPL version 2 or newer`.
- The files `inst/doc/src/mainmatter/acnuc_sockets.rnw .tex` with non-portable file names were changed to `acnucsocket.rnw` and `acnucsocket.tex`.

release 1.1-3

- There is a new chapter to explain how to set up a local ACNUC server on Unix-like platforms.
- New dataset `m16j` to make a GC skew plot as in [19].
- New dataset `waterabs` giving the absorption of light by water. This dataset was compiled by Palmeira [24] from [18, 27].
- Generic functions `getAnnot()`, `getFrag()`, `getKeyword()`, `getLength()`, `getLocation()`, `getName()`, `getSequence()` and `getTrans()` have gained methods to handle objects from class `list` and `qaw`.

- Functions `getAttributsocket()` and `getNumber.socket()` are now deprecated, a warning is issued.
- There is a new appendix in which all the examples protected by a `dontrun` statment are forced to be executed.
- Function `read.fasta()` now supports comment lines starting by a semicolon character in FASTA files. An example of such a file is provided in `sequences/legacy.fasta`. The argument `File` is now deprecated. There is a new argument `seqonly` to import just the sequences without names, annotations and coercion attempts. There is a new argument `strip.desc` to remove the leading '>' character in annotations (as in function `readFASTA` from the Biostrings package [23]). The FASTA file example `someORF.fsa` from Biostrings is also added for comparisons.
- Function `GC()` has gained a new argument `NA.GC` defaulting to `NA` to say what should be returned when the GC content cannot be computed from data (for instance with a sequence like NNNNNNNNNNN). The argument `oldGC` is now deprecated and a warning is issued. Functions `GC1()`, `GC2()`, `GC3()` are now simple wrappers for the more general `GCpos()` function. The new argument `frame` allows to take the frame into account for CDS.
- Function `read.alignment()` has gained a new argument `forceToLower` defaulting to TRUE to force lower case in the character of the sequence (this is for a smoother interaction with the package `ape`). The argument `File` is now deprecated and a warning is issued when used instead of `file`. The example in the function `kaks()` has been corrected to avoid this warning when reading the example files.
- New low level utility function `acnucclose()` and `quitacnuc()` to close an ACNUC server. These functions are called by `closebank()` so that a simple call to it should be enough.
- New low level utility function `clientid()` to send the client ID to an ACNUC server.
- New low level utility function `countfreelists()` to get the number of free lists available in an ACNUC server.
- New low level utility function `knowndbs()` and its shortcut `kdb()` to get a description of databases known by an ACNUC server.
- New low level utility function `autosocket()` to get the socket connection to the last opened ACNUC database.
- New function `countsubseqs()` to get the number of subsequences in an ACNUC list.

- New function `savelist()` to save sequence names or accession numbers from an ACNUC list into a local file.
- New function `ghelp()` to get help from an ACNUC server.
- New function `modifylist()` to modify a previously existing ACNUC list by selecting sequences either by length, either by date, either for the presence of a given string in annotations.
- New low level function `getlistate()` to ask for information about an ACNUC list.
- New low level function `setlistname()` to set the name of a list from an ACNUC server.
- New function `residuecount()` to count the total number of residues (nucleotides or aminoacids) in all sequences of an ACNUC list of specified rank.
- New function `isenum()` and its shortcut `isn()` to get the ACNUC number of a sequence from its name or accession number.
- New function `prettyseq()` to get a text representation of a sequence from an ACNUC server.
- New function `gfrag()` to extract sequence identified by name or by number from an ACNUC server.
- The details of the socket connection are no more stored in the slot `socket` for objects of class `seqAcnucWeb`: this slot is now deleted. As a consequence, the argument `socket` in function `as.SeqAcnucWeb()` has been removed and there is now a new argument `socket = "auto"` in functions `getAnnot()`, `getFrag()`, `geyKeyword()`, `getLocation()`, and `getSequence()`. The default value `"auto"` means that the details of the socket connection are taken automatically when necessary from the last opened bank. The size of local lists of sequences is reduced by about a third now as compared to the previous version.
- New function `print.seqAcnucWeb()` to print objects from class `seqAcnucWeb`.
- Internal function `parser.socket()` has been optimized and is about four times faster now. This decreases the time needed by the `query()` function.

release 1.1-2 (27-SEP-2007)

- New function `trimSpace()` to remove leading and trailing spaces in string vectors.
- Function `splitseq()` is no more based on `substring()`, it is now more efficient for long sequences.

- A sanity check test was added in the documentation file for the function `syncodons()`.
- The way this manual is produced is now documented in the `doc/src/template/` folder.
- A bug in function `oriloc()` was reported on 23 Jul 2007 by Michael KUBE: using directly genBank files was no more possible. The culprit was `gbk2g2()` that turns genBank files into glimmer files version 2 when `oriloc()` default is to use version 3 files. The `glimmer.version` argument is now forced to 2 when working with genBank files to fix this problem.
- Function `zscore()` has now a new argument `exact` (which is only effective for the option `model = base`). This argument, when set to `TRUE` allows for the exact analytical computation of the zscore under this model, instead of the approximation for large sequences. It is set to `FALSE` by default for backward compatibility.

release 1.1-1 (20-JUL-2007)

- A bug was reported by Sylvain MOUSSET on 14 Jul 2007 in function `dist.alignment()`: when called with sequences in lower case letters, some sequences were modified. This should no more be the case:

```
ali <- list(nb=4, nam=c("speciesA", "speciesB", "speciesC", "speciesD"),
seq=c("ACGT","acgt","ACGT","ACGT"))
class(ali) <- "alignment"
print(ali$seq)
```

```
[1] "ACGT" "acgt" "ACGT" "ACGT"
```

```
print(dist.alignment(ali))
```

```
         speciesA speciesB speciesC
speciesB        0
speciesC        0        0
speciesD        0        0        0
```

```
print(ali$seq)
```

```
[1] "ACGT" "acgt" "ACGT" "ACGT"
```

- The CITATION file has been updated so that now `citation("seqinr")` returns the full complete reference for the package seqinR.
- Non ASCII characters in documentation (*.Rd) files have been removed. Declaration of the encoding as latin1 when necessary is now present. The updated documentation files are: `dinucl.Rd, gb2fasta.Rd, get.ncbi.Rd, lseqinr.Rd, n2s.Rd, prochlo.Rd, s2c.Rd, SeqAcnucWeb.Rd, SeqFrag.Rd, toyaa.Rd, words.pos.Rd, words.Rd, zscore.Rd`.
- Function `GC()` and by propagation functions `GC1()`, `GC2()` and `GC3()` have gained a new argument `oldGC` allowing to compute the G+C content as in releases up to 1.0-6 included. The code has been also modified to avoid divisions by zero with very small sequences.

- New function `rot13()` that returns the ROT-13 encoding of a string of characters.

1.0 series

release 1.0-7 (19-APR-2007)

- A new *experimental* function `extractseqs()` to download sequences thru zlib compressed sockets from an ACNUC server is released. Preliminary tests suggest that working with about 100,000 CDS is possible with a home ADSL connection. See the manual for some `system.time()` examples.

- As pointed by e-mail on 16 Nov 2006 by Emmanuel PRESTAT the URL used in `dia.bactgensize()` was no more available, this has been fixed in the current version.

- As pointed by e-mail on 16 Nov 2006 by Guy PERRIÈRE, the function `oriloc()` was no more compatible with glimmer[2] 3.0 outputs. The function has gained a new argument `glimmer.version` defaulting to 3, but the value 2 is still functional for backward compatibility with old glimmer outputs.

- As pointed by e-mail on 24 Oct 2006 by Lionel GUY (`http://pbil.univ-lyon1.fr/seqinr/seqinrhtmlannuel/03/0089.html`) there was no default value for the `as.string` argument in the `getSequence.SeqFastadna()`. A default `FALSE` value is now present for backward compatibility with older code.

- New utility vectorized function `stresc()` to escape LaTeX special characters present in a string.

- New low level function `readsmj()` available.

- A new function `readfirstrec()` to get the record count of the specified ACNUC index file is now available.

- Function `getType()` called without arguments will now use the default ACNUC database to return available subsequence types.

- Function `read.alignment()` now also accepts `file` in addition to `File` as argument.

- A new function `rearranged.oriloc()` is available. This method, based on `oriloc()`, can be used to detect the effect of the replication mechanism on DNA base composition asymmetry, in prokaryotic chromosomes.

[2] Glimmer is a program to predict coding sequences in microbial genomes [29, 4].

- New function `extract.breakpoints()`, used to extract breakpoints in rearranged nucleotide skews. This function uses the `segmented` package to define the position of the breakpoints.
- New function `draw.rearranged.oriloc()` available, to plot nucleotide skews on artificially rearranged prokaryotic chromosomes.
- New function `gbk2g2.euk()` available. Similarly to `gbk2g2()`, this function extracts the coding sequence annotations from a GenBank format file. This function is specifically designed for eukaryotic sequences, *i.e.* with introns. The output file will contain the coordinates of the exons, along with the name of the CDS to which they belong.
- After an e-mail by Marcelo Bertalan on 26 Mar 2007, a bug in `oriloc()` when the `gbk` argument was `NULL` was found and fixed by Anamaria Necşulea.
- Functions `translate()` and `getTrans()` have gained a new argument `NAstring` to represent untranslatable amino- acids, defaulting to character "X".
- There was a typo for the total number of printed bases in the ACNUC books [6, 7] : 474,439 should be 526,506.
- Function `invers()` has been deleted.
- Functions `translate()`, `getTrans()` and `comp()` have gained a new argument `ambiguous` defaulting to FALSE allowing to handle ambiguous bases. If TRUE, ambiguous bases are taken into account so that for instance GGN is translated to Gly in the standard genetic code.
- New function `amb()` to return the list of nucleotide matching a given IUPAC nucleotide symbol.
- Function `count()` has gained a new argument `alphabet` so that oligopeptides counts are now possible. Thanks to Gabriel Valiente for this suggestion. The functions `zscore()`, `rho()` and `summary.SeqFastadna()` have also an argument `alphabet` which is forwarded to `count()`.

release 1.0-6 (06-SEP-2006)

Release 1.0-6 is a minor release to fix a problem found and solved by Kurt Hornik (namely a change from `SET_ELEMENT` to `SET_STRING_ELT` in C code for `s2c()` in file `util.c`). The few changes are as follows.

- More typographical option for the output LaTeX table of `tablecode()` are now available to outline deviations from the standard genetic code (see example in the appendix "genetic codes" of the manual).

- A new dataset `aaindex` extracted from the aaindex database [12, 34, 21] is now available. It contains a list of 544 physicochemical and biological properties for the 20 amino-acids
- The default value for argument `dia` is now `FALSE` in function `tablecode()`.
- The example code for `data(chargaff)` has been changed.

release 1.0-5 (21-JUL-2006)

- A new function `dotPlot()` is now available.
- A new function `crelistfromclientdata()` is now available to create a list on the server from a local file of sequence names, sequence accession numbers, species names, or keywords names.
- A new function `pmw()` to compute the molecular weight of a protein is now available.
- A new function `reverse.align()` contributed by Anamaria Necşulea is now available to align CDS at the protein level and then reverse translate this at the nucleic acid level from a `clustalw` output. This can be done on the fly if `clustalw` is available on your platform.
- An undocumented behavior was reported by Guy Perrière for `uco()` when computing RSCU on sequences where an amino-acid is missing. There is now a new argument `NA.rscu` that allows the user to force the missing values to his favorite magic value.
- There was a bug in `read.fasta()`: some sequence names were truncated, this is now fixed (thanks to Marcus G. Daniels for pointing this). In order to be more consistent with standard functions such as `read.table()` or `scan()`, the file argument starts now with a lower case letter (`file`) in function `read.fasta()`, but the old-style `File` is still functional for forward-compatibility. There is a new logical argument in `read.fasta()` named `as.string` to allow sequences to be returned as strings instead of vector of single characters. The automatic conversion of DNA sequences into lower case letters can now be disabled with the new logical argument `forceDNAtolower`. It is also possible to disable the automatic attributes settings with the new logical argument `set.attributes`.
- A new function `write.fasta()` is now available.
- The function `kaks()` now forces character in sequences to upper case. This default behavior can be neutralized in order to save time by setting the argument `forceUpperCase` to `FALSE`.

release 1.0-4 (30-MAR-2006)

- The scaling factor $n_{\bullet\bullet}$ was missing in equation **??**.
- The files `louse.fasta`, `louse.names`, `gopher.fasta`, `gopher.names` and `ortho.fasta` that were used for examples in the previous version of this document are no more downloaded from the internet since they are now distributed in the `sequences/` folder of the package.
- An example of synonymous and non synonymous codon usage analysis was added to the vignette along with two toy data sets (`toyaa` and `toycodon`).
- A FAQ section was added to the vignette.
- A bug in `getAnnot()` when the number of lines was zero is now fixed.
- There is now a new argument, `latexfile`, in `tablecode()` to export genetic codes tables in a LaTeX document, for instance table **??** and table **??** here.
- There is now a new argument, `freq`, in `count()` to compute word frequencies instead of counts.
- Function `splitseq()` has been entirely rewritten to improve speed.
- Functions computing the G+C content: `GC(), GC1(), GC2(), GC3()` were rewritten to improve speed, and their document files were merged to facilitate usage.
- The following new functions have been added:
  - `syncodons()` returns all synonymous codons for a given codon. Argument `numcode` specifies the desired genetic code.
  - `ucoweight()` returns codon usage bias on a sequence as the number of synonymous codons present in the sequence for each amino acid.
  - `synsequence()` generates a random coding sequence which is synonymous to a given sequence and has a chosen codon usage bias.
  - `permutation()` generates a new sequence from a given sequence, while maintaining some constraints from the given sequence such as nucleotide frequency, codon usage bias, ...
  - `rho()` computes the rho statistic on dinucleotides as defined in [10].
  - `zscore()` computes the zscore statistic on dinucleotides as defined in [25].
- Two datasets (`dinucl` and `prochlo`) were added to illustrate these new functions.

release 1.0-3 (26-JUL-2005)

- The new package maintainer is Dr. Simon PENEL, PhD, who has now a fixed position in the laboratory that issued **seqinR** (`penel@biomserv.univ-lyon1.fr`). Delphine CHARIF was successful too to get a fixed position in the same lab, with now a different research task (but who knows?). Thanks to the close vicinity of our pioneering maintainers the transition was sweet. The DESCRIPTION file of the **seqinR** package has been updated to take this into account.

- The reference paper for the package is now *in press*. We do not have the full reference for now, you may use `citation("seqinr")` to check if it is complete now:

```
 citation("seqinr")
To cite seqinr in publications use:

  Charif, D. and Lobry, J.R. (2007)

Une entrée BibTeX pour les utilisateurs LaTeX est

  @InCollection{,
    author = {D. Charif and J.R. Lobry},
    title = {Seqin{R} 1.0-2: a contributed package to the {R} project for statistical computing devoted to
    booktitle = {Structural approaches to sequence evolution: Molecules, networks, populations},
    year = {2007},
    editor = {U. Bastolla and M. Porto and H.E. Roman and M. Vendruscolo},
    series = {Biological and Medical Physics, Biomedical Engineering},
    pages = {207-232},
    address = {New York},
    publisher = {Springer Verlag},
    note = {{ISBN :} 978-3-540-35305-8},
  }
```

- There was a bug when sending a `gfrag` request to the server for long (Mb range) sequences. The length argument was converted to scientific notations that are not understand by the server. This is now corrected and should work up the the Gb scale.

- The `query()` function has been improved by de-looping list element info request, there are now download at once which is much more efficient. For example, a query from a researcher-home ADSL connection with a list with about 1000 elements was 60 seconds and is now only 4 seconds (*i.e.* 15 times faster now).

- A new parameter `virtual` has been added to `query()` so that long lists can stay on the server without trying to download them automatically. A query like `query(s$socket,"allcds","t=cds", virtual = TRUE)` is now possible.

- Relevant genetic codes and frames are now automatically propagated.

- **seqinR** sends now its name and version number to the server.

- Strict control on ambiguous DNA base alphabet has been relaxed.

- Default value for parameter `invisible` of function `query()` is now `TRUE`.

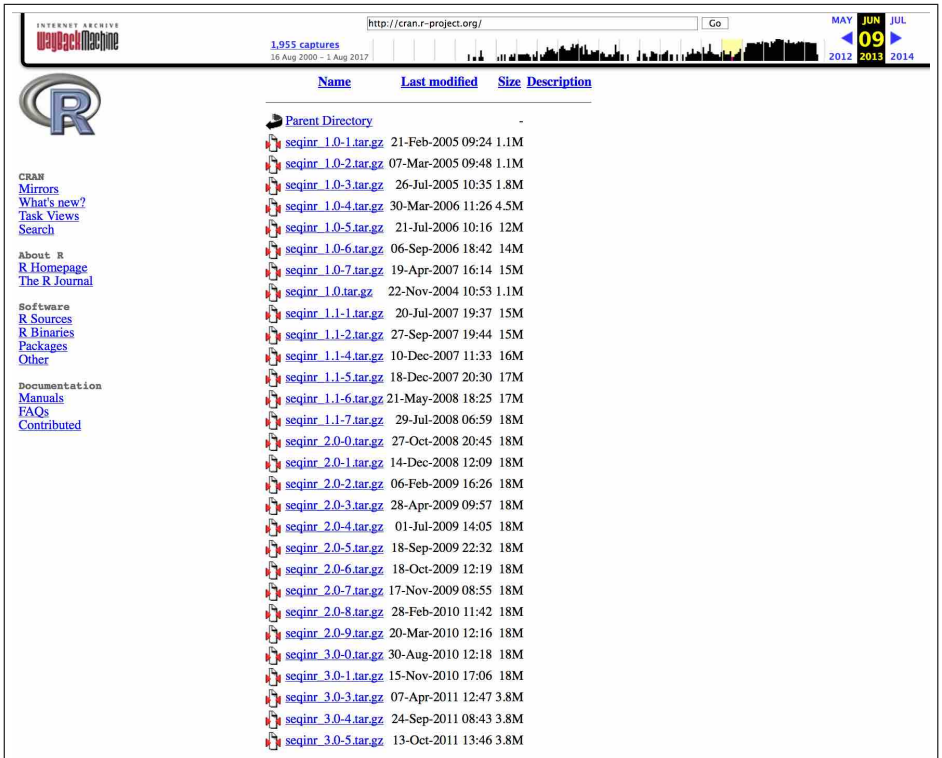


Figure 1: Screenshot from the internet archive wayback machine showing the `seqinR` archive listing on CRAN on 09-JUN-2013. Releases from 1.0-7 to 3.1-1, included, are lost for ever in the present CRAN archive for `seqinR`.

release 1.0-2 (07 MAR 2005)

The citation [3] for the `seqinR` package is based on this release.

release 1.0-1 (21 FEV 2005)

There were no release notes for the `seqinR` package at that time.

release 1.0 (22 NOV 2004)

This is the only release that does not respect the `X.Y-Z` scheme for release numbers. There were no release notes for the `seqinR` package at that time. The first public presentation of `seqinR` was a seminar (02-JUL-2003, Lausanne University, Swiss) and the first public release on the CRAN was in october 2004.

Session Informations

This part was compiled under the following R 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 10.12.6`
- 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 1.7-8, ape 4.1, grImport 0.9-0, MASS 7.3-47, seqinr 3.4-6, 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:

- R compilation time was: Sat Feb 24 11:46:24 2018
- LaTeX compilation time was: February 24, 2018

References

[1] W. Bar, B. Brinkmann, P. Lincoln, W.R. Mayr, and U. Rossi. DNA recommendations. 1994 report concerning further recommendations of the DNA commission of the ISFH regarding PCR-based polymorphisms in STR (short tandem repeat) systems. *Int. J. Leg. Med.*, 107:159–160, 1994.

[2] A. Bernal, U. Ear, and N. Kyrpides. Genomes online database (GOLD): a monitor of genome projects world-wide. *Nucleic Acids Research*, 29:126–127, 2001.

[3] 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.

[4] A.L. Delcher, D. Harmon, S. Kasif, O. White, and S.L. Salzberg. Improved microbial gene identification with GLIMMER. *Nucleic Acids Research*, 27:4636–4641, 1999.

[5] N. Galtier and J.R. Lobry. Relationships between genomic G+C content, RNA secondary structures, and optimal growth temperature in prokaryotes. *Journal of Molecular Evolution*, 44:632–635, 1997.

[6] 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.

[7] 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.

[8] C. Gautier, M. Gouy, and S. Louail. Non-parametric statistics for nucleic acid sequence study. *Biochimie*, 67:449–453, 1985.

[9] M. Gouy, F. Milleret, C. Mugnier, M. Jacobzone, and C. Gautier. ACNUC: a nucleic acid sequence data base and analysis system. *Nucleic Acids Res.*, 12:121–127, 1984.

[10] S. Karlin and V. Brendel. Chance and statistical significance in protein and DNA sequence analysis. *Science*, 257:39–49, 1992.

[11] S. Karlin and L.R. Cardon. Computational DNA sequence analysis. *Annual Review of Microbiology*, 48:619–654, 1994.

[12] S. Kawashima and M. Kanehisa. AAindex: amino acid index database. *Nucleic Acids Res.*, 28:374–374, 2000.

[13] J. Kiraga. *Analysis and computer simulations of variability of isoelectric point of proteins in the proteomes.* PhD thesis, University of Wrocław, 2008.

[14] J. Krawczyk, A. Goesmann, R. Nolte, M. Werber, and B. Weisshaar. Trace2PS and FSA2PS: two software toolkits for converting trace and fsa files to PostScript format. *Source Code for Biology and Medicine*, 4:4, 2009.

[15] N.C. Kyrpides. Genomes online database (GOLD 1.0): a monitor of complete and ongoing genome projects world-wide. *Bioinformatics*, 15:773–774, 1999.

[16] K. Liolios, K. Mavrommatis, N. Tavernarakis, and N.C. Kyrpides. The genomes on line database (GOLD) in 2007: status of genomic and metagenomic projects and their associated metadata. *Nucleic Acids Research*, in press:D000–D000, 2008.

[17] K. Liolios, N. Tavernarakis, P. Hugenholtz, and N.C. Kyrpides. The genomes on line database (GOLD) v.2: a monitor of genome projects worldwide. *Nucleic Acids Research*, 34:D332–D334, 2006.

[18] R.A. Litjens, T.I. Quickenden, and C.G. Freeman. Visible and near-ultraviolet absorption spectrum of liquid water. *Applied Optics*, 38:1216–1223, 1999.

[19] J.R. Lobry. Asymmetric substitution patterns in the two DNA strands of bacteria. *Molecular Biology and Evolution*, 13:660–665, 1996.

[20] J.R. Lobry and D. Chessel. Internal correspondence analysis of codon and amino-acid usage in thermophilic bacteria. *Journal of Applied Genetics*, 44:235–261, 2003.

[21] K. Nakai, A. Kidera, and M. Kanehisa. Cluster analysis of amino acid indices for prediction of protein structure and function. *Protein Eng.*, 2:93–100, 1988.

[22] H. Naya, H. Romero, A. Zavala, B. Alvarez, and H. Musto. Aerobiosis increases the genomic guanine plus cytosine content (GC%) in prokaryotes. *Journal of Molecular Evolution*, 55:260–264, 2002.

[23] H. Pages, R. Gentleman, and S. DebRoy. *Biostrings: String objects representing biological sequences, and matching algorithms*, 2007. R package version 2.6.4.

[24] L. Palmeira. *Analyse et modélisation des dépendances entre sites voisins dans l'évolution des séquences d'ADN.* PhD thesis, Université Claude Bernard - Lyon I, 2007.

[25] L. Palmeira, L. Guéguen, and J.R. Lobry. UV-targeted dinucleotides are not depleted in light-exposed prokaryotic genomes. *Molecular Biology and Evolution*, 23:2214–2219, 2006.

[26] J.F. Peden. *Analysis of codon usage*. PhD thesis, University of Nottingham, 1999.

[27] T.I. Quickenden and J.A. Irvin. The ultraviolet absorption spectrum of liquid water. *The Journal of Chemical Physics*, 72:4416–4428, 1980.

[28] Trina E. Roberts. *ComPairWise: Compare phylogenetic or population genetic data alignments*, 2007. R package version 1.01.

[29] S.L. Salzberg, A.L. Delcher, S. Kasif, and O. White. Microbial gene identification using interpolated Markov models. *Nucleic Acids Research*, 26:544–548, 1998.

[30] P.M. Sharp and E. Cowe. Synonymous codon usage in *Saccharomyces cerevisiae*. *Yeast*, 7:657–678, 1991.

[31] P.M. Sharp and W.-H. Li. The codon adaptation index - a measure of directional synonymous codon usage bias, and its potential applications. *Nucleic Acids Research*, 15:1281–1295, 1987.

[32] D.C. Shields and P.M. Sharp. Synonymous codon usage in *Bacillus subtilis* reflects both translational selection and mutational biases. *Nucleic Acids Research*, 15:8023–8040, 1987.

[33] H. Suzuki, C.J. Brown, L.J. Forney, and E. Top. Comparison of correspondence analysis methods for synonymous codon usage in bacteria. *DNA Research*, 15:357–365, 2008.

[34] K. Tomii and M. Kanehisa. Analysis of amino acid indices and mutation matrices for sequence comparison and structure prediction of proteins. *Protein Eng.*, 9:27–36, 1996.