R: analiza sekvenc

A. Blejec

December 5, 2012

Contents

1 Paket seqinr

V paketu **seqinr** najdemo funkcije za manipulacijo sekvenc (Charif and Lobry, 2007). Paket je dobro dokumentiran, dokumentacija je dostopna na R-Forgev SeqinR 2.0-7.

```
> library(seqinr)
> if(interactive()) library(help=seqinr)
```

Nekatere funkcije paketa **seqinr**

```
> tablecode()
```

Genetic code 1 : standard							
T T T	Phe	T CT	Ser	T A T	Tyr	T GT	Cys
T T C	Phe	T CC	Ser	T A C	Tyr	T GC	Cys
T T A	Leu	T CA	Ser	T A A	Stp	T GA	Stp
T T G	Leu	T CG	Ser	T A G	Stp	T GG	Trp
CT T	Leu	CCT	Pro	CA T	His	CGT	Arg
CT C	Leu	CCC	Pro	CA C	His	CGC	Arg
CT A	Leu	CCA	Pro	CA A	Gln	CGA	Arg
CT G	Leu	CCG	Pro	CA G	Gln	CGG	Arg
AT T	Ile	ACT	Thr	AA T	Asn	AGT	Ser
AT C	Ile	ACC	Thr	AA C	Asn	AGC	Ser
AT A	Ile	ACA	Thr	AA A	Lys	AGA	Arg
AT G	Met	ACG	Thr	AA G	Lys	AGG	Arg
GT T	Val	GCT	Ala	GA T	Asp	GGT	Gly
GT C	Val	GCC	Ala	GA C	Asp	GGC	Gly
GT A	Val	GCA	Ala	GA A	Glu	GGA	Gly
GT G	Val	GCG	Ala	GA G	Glu	GGG	Gly

```
> tablecode(latexfile="tablecode.tex")
```

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
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 1: Genetic code number 1: standard.

2 ACNUC baze

Opis s spletne strani o ACNUC:

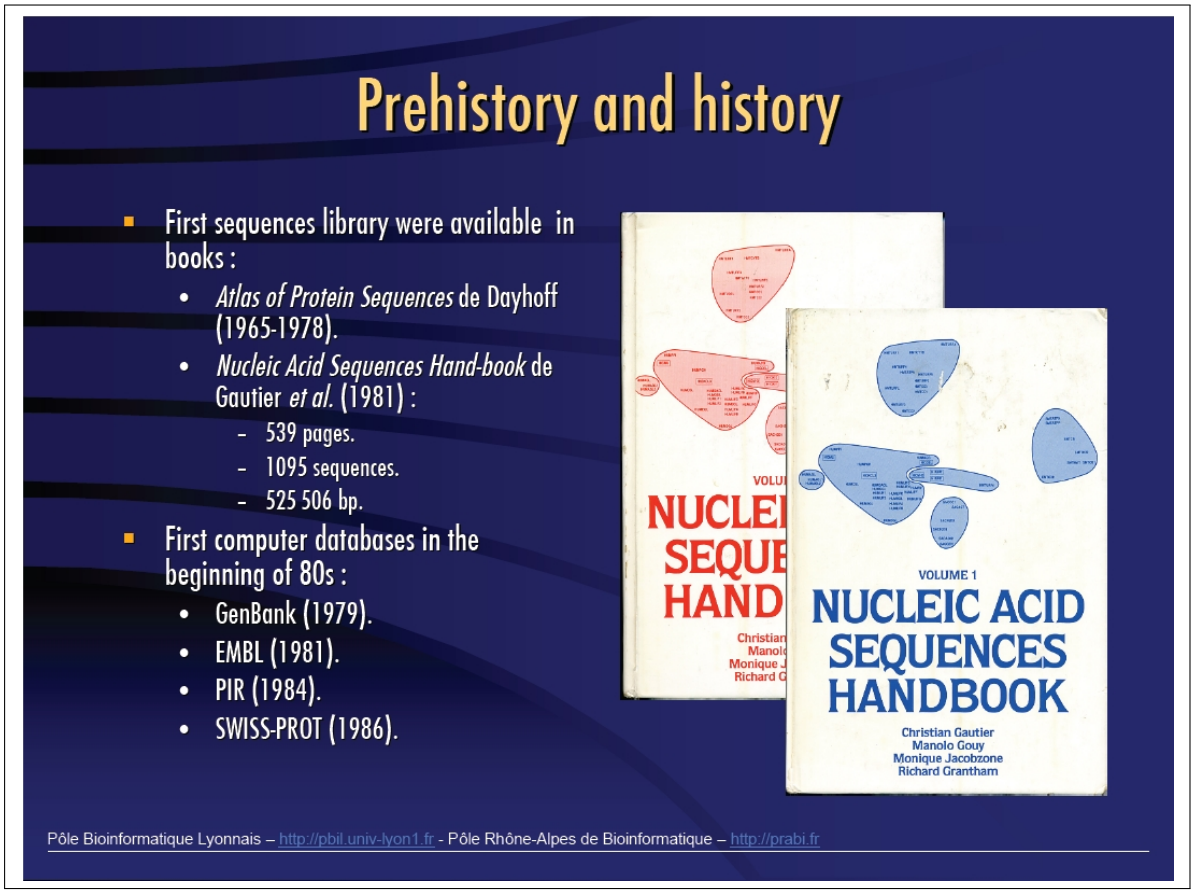
> ACNUC is a retrieval system for the nucleotide and protein sequence databases GenBank, EMBL, UniProt/SWISS-PROT or NBRF-PIR, and for many other databases following the same formats.

Kratica ACNUC je izpeljana kot okrajšava francoskega izraza ACides NUCleiques (http://pbil. univ-lyon1.fr/databases/acnuc/acnuc.html)

Interaktiven dostop do ACNUC baz omogoča tudi program raa_query_win.exe
.

Katere baze podatkov so dostopne?

```
> choosebank()
 [1] "genbank"        "embl"           "emblwgs"        "swissprot"
 [5] "ensembl"        "hogenom"        "hogenomdna"     "hovergendna"
 [9] "hovergen"       "hogenom5"       "hogenom5dna"    "hogenom4"
[13] "hogenom4dna"    "homolens"       "homolensdna"    "hobacnucl"
[17] "hobacprot"      "phever2"        "phever2dna"     "refseq"
[21] "greviews"       "bacterial"      "protozoan"      "ensbacteria"
[25] "ensprotists"    "ensfungi"       "ensmetazoa"     "ensplants"
[29] "mito"           "polymorphix"    "emglib"         "taxobacgen"
[33] "refseqViruses"
```



../clp/capture-ACNUC.jpg

http://www.genouest.org/documents/Evenements/200710_-ReNabiWorkshop/20071022_SimonPenel.pdf

Prvi dve knjigi sta debeli kakih 4.5cm in sta vsebovali 526506 baznih parov. Kako dolgo bi bilo danes v tiskani obliki vse kar je v bazi genebank? S pomočjo R in posegom v bazo genebank lahko ekstrahiramo podatke iz opisa.

```
> acnucbooksize <- 4.5
> acnucbp <- 526506
> mybank <- choosebank("genbank")
> closebank()
> mybank$details
```

```
[1] "             ****     ACNUC Data Base Content      ****
[2] "        GenBank Release 192 (15 October 2012) Last Updated: Dec  5, 2012"
[3] "148,164,066,733 bases; 160,964,295 sequences; 13,564,090 subseqs; 711,487
[4] "Software by M. Gouy, Lab. Biometrie et Biologie Evolutive, Universite Lyo
```

```
> bpbk <- unlist(strsplit(mybank$details[3], split = " "))[1]
> bpbk
```

```
[1] "148,164,066,733"
```

```
> bpbk <- as.numeric(paste(unlist(strsplit(bpbk, split = ",")),
+ collapse = ""))
> widthcm <- acnucbooksize * bpbk/acnucbp
> (widthkm <- widthcm/10^5)
```

```
[1] 12.66345
```

Če bi vse skupaj natisnili, bi bilo knjig za 12.7km!!
Leta 2011 jih je bilo za 11.3km.

2.1 Funkcije paketa seqinr

```
> lseqinr()
  [1] "a"                          "aaa"
  [3] "AAstat"                     "acnucclose"
  [5] "acnucopen"                  "al2bp"
  [7] "alllistranks"               "alr"
  [9] "amb"                        "as.alignment"
 [11] "as.matrix.alignment"        "as.SeqAcnucWeb"
 [13] "as.SeqFastaAA"              "as.SeqFastadna"
 [15] "as.SeqFrag"                 "autosocket"
 [17] "baselineabif"               "bma"
 [19] "c2s"                        "cai"
 [21] "cfl"                        "choosebank"
 [23] "circle"                     "clfcd"
 [25] "clientid"                   "closebank"
 [27] "col2alpha"                  "comp"
 [29] "computePI"                  "con"
 [31] "consensus"                  "count"
 [33] "countfreelists"             "countsubseqs"
 [35] "crelistfromclientdata"      "css"
 [37] "dia.bactgensize"            "dia.db.growth"
 [39] "dist.alignment"             "dotchart.uco"
 [41] "dotPlot"                    "draw.oriloc"
 [43] "draw.rearranged.oriloc"     "draw.recstat"
 [45] "EXP"                        "exseq"
 [47] "extract.breakpoints"        "extractseqs"
 [49] "fastacc"                    "gb2fasta"
 [51] "gbk2g2"                     "gbk2g2.euk"
 [53] "GC"                         "GC1"
 [55] "GC2"                        "GC3"
 [57] "GCpos"                      "get.db.growth"
 [59] "get.ncbi"                   "getAnnot"
 [61] "getAnnot.default"           "getAnnot.list"
 [63] "getAnnot.logical"           "getAnnot.qaw"
 [65] "getAnnot.SeqAcnucWeb"       "getAnnot.SeqFastaAA"
 [67] "getAnnot.SeqFastadna"       "getAttributsocket"
 [69] "getFrag"                    "getFrag.character"
 [71] "getFrag.default"            "getFrag.list"
 [73] "getFrag.logical"            "getFrag.qaw"
 [75] "getFrag.SeqAcnucWeb"        "getFrag.SeqFastaAA"
 [77] "getFrag.SeqFastadna"        "getFrag.SeqFrag"
 [79] "getKeyword"                 "getKeyword.default"
 [81] "getKeyword.list"            "getKeyword.logical"
 [83] "getKeyword.qaw"             "getKeyword.SeqAcnucWeb"
 [85] "getLength"                  "getLength.character"
 [87] "getLength.default"          "getLength.list"
 [89] "getLength.logical"          "getLength.qaw"
 [91] "getLength.SeqAcnucWeb"      "getLength.SeqFastaAA"
 [93] "getLength.SeqFastadna"      "getLength.SeqFrag"
 [95] "getlistrank"                "getliststate"
 [97] "getLocation"                "getLocation.default"
 [99] "getLocation.list"           "getLocation.logical"
[101] "getLocation.qaw"            "getLocation.SeqAcnucWeb"
```

```
[103] "getName"                  "getName.default"
[105] "getName.list"             "getName.logical"
[107] "getName.qaw"              "getName.SeqAcnucWeb"
[109] "getName.SeqFastaAA"       "getName.SeqFastadna"
[111] "getName.SeqFrag"          "getNumber.socket"
[113] "getSequence"              "getSequence.character"
[115] "getSequence.default"      "getSequence.list"
[117] "getSequence.logical"      "getSequence.qaw"
[119] "getSequence.SeqAcnucWeb" "getSequence.SeqFastaAA"
[121] "getSequence.SeqFastadna" "getSequence.SeqFrag"
[123] "getTrans"                 "getTrans.character"
[125] "getTrans.default"         "getTrans.list"
[127] "getTrans.logical"         "getTrans.qaw"
[129] "getTrans.SeqAcnucWeb"     "getTrans.SeqFastadna"
[131] "getTrans.SeqFrag"         "getType"
[133] "gfrag"                    "ghelp"
[135] "gln"                      "glr"
[137] "gls"                      "is.SeqAcnucWeb"
[139] "is.SeqFastaAA"            "is.SeqFastadna"
[141] "is.SeqFrag"               "isenum"
[143] "isn"                      "kaks"
[145] "kdb"                      "knowndbs"
[147] "lseqinr"                  "modifylist"
[149] "move"                     "mv"
[151] "n2s"                      "ncbi.fna.url"
[153] "ncbi.gbk.url"             "ncbi.ptt.url"
[155] "ncbi.stats"               "oriloc"
[157] "parser.socket"            "peakabif"
[159] "permutation"              "pga"
[161] "plot.SeqAcnucWeb"         "plotabif"
[163] "plotladder"               "plotPanels"
[165] "pmw"                      "prepgetannots"
[167] "prettyseq"                "print.qaw"
[169] "print.SeqAcnucWeb"        "query"
[171] "quitacnuc"                "read.abif"
[173] "read.alignment"           "read.fasta"
[175] "readBins"                 "readfirstrec"
[177] "readPanels"               "readsmj"
[179] "rearranged.oriloc"        "recstat"
[181] "residuecount"             "reverse.align"
[183] "rho"                      "rot13"
[185] "s2c"                      "s2n"
[187] "savelist"                 "SEQINR.UTIL"
[189] "setlistname"              "splitseq"
[191] "stresc"                   "stutterabif"
[193] "summary.SeqFastaAA"       "summary.SeqFastadna"
[195] "swap"                     "syncodons"
[197] "synsequence"              "tablecode"
[199] "test.co.recstat"          "test.li.recstat"
[201] "translate"                "trimSpace"
[203] "uco"                      "ucoweight"
[205] "where.is.this.acc"        "words"
[207] "words.pos"                "write.fasta"
```

```
[209] "zscore"
>
```

3 Primerjava zaporedij

3.1 Iskanje podobnosti zaporedij

Spremeba zaporedja v vektor znakov

```
> (seq1 <-"GGGATCACG")
[1] "GGGATCACG"
> (v <- s2c(seq1))
[1] "G" "G" "G" "A" "T" "C" "A" "C" "G"
```

Primerjava lege nukleotidov

```
> (P <- outer(v,v,"==")+0)
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
 [1,]    1    1    1    0    0    0    0    0    1
 [2,]    1    1    1    0    0    0    0    0    1
 [3,]    1    1    1    0    0    0    0    0    1
 [4,]    0    0    0    1    0    0    1    0    0
 [5,]    0    0    0    0    1    0    0    0    0
 [6,]    0    0    0    0    0    1    0    1    0
 [7,]    0    0    0    1    0    0    1    0    0
 [8,]    0    0    0    0    0    1    0    1    0
 [9,]    1    1    1    0    0    0    0    0    1
```

Naredimo preglednejšo z dodatkom imen. Ničle so izpisane manj vidno.

```
> dimnames(P) <- list(v,v)
> print.table(P,zero.print=".")
  G G G A T C A C G
G 1 1 1 . . . . . 1
G 1 1 1 . . . . . 1
G 1 1 1 . . . . . 1
A . . . 1 . . 1 . .
T . . . . 1 . . . .
C . . . . . 1 . 1 .
A . . . 1 . . 1 . .
C . . . . . 1 . 1 .
G 1 1 1 . . . . . 1
```

4 dot-plot

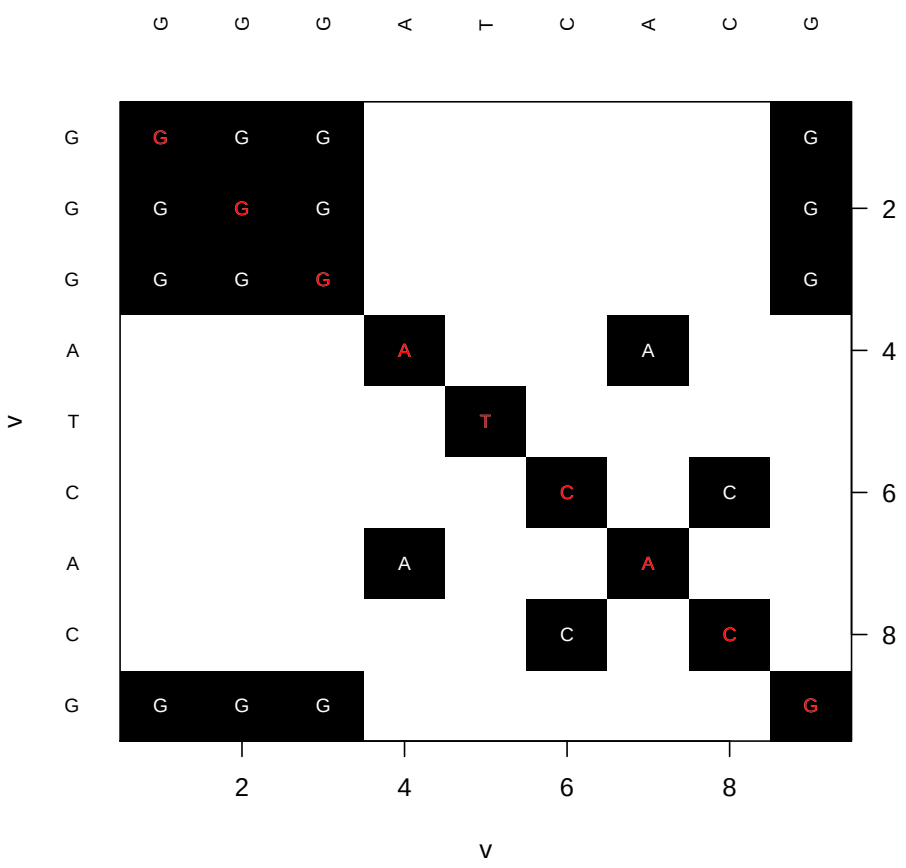
Dot plot je v analize sekvenc vpeljal Maizel (**?**).

```
> dotPlot <-
+ function (seq1, seq2, wsize = 1, wstep = 1, nmatch = 1, col = c("white",
+     "black"), xlab = deparse(substitute(seq1)), ylab = deparse(substitute(se
+    label=TRUE,
+     ...)
+ {
+     if (nchar(seq1[1]) > 1)
+         stop("seq1 should be provided as a vector of single chars")
+     if (nchar(seq2[1]) > 1)
+         stop("seq2 should be provided as a vector of single chars")
+     if (wsize < 1)
+         stop("non allowed value for wsize")
+     if (wstep < 1)
+         stop("non allowed value for wstep")
+     if (nmatch < 1)
+         stop("non allowed value for nmatch")
+     if (nmatch > wsize)
+         stop("nmatch > wsize is not allowed")
+     mkwin <- function(seq, wsize, wstep) {
+         sapply(seq(from = 1, to = length(seq) - wsize + 1, by = wstep),
+             function(i) c2s(seq[i:(i + wsize - 1)]))
+     }
+     wseq1 <- mkwin(seq1, wsize, wstep)
+     wseq2 <- mkwin(seq2, wsize, wstep)
+     if (nmatch == wsize) {
+         xy <- outer(wseq1, rev(wseq2), "==")
+     }
+     else {
+         "%==%" <- function(x, y) colSums(sapply(x, s2c) == sapply(y,
+             s2c)) >= nmatch
+         xy <- outer(wseq1, rev(wseq2), "%==%")
+     }
+     image(x = seq(from = 1, to = length(wseq1), length = length(wseq1)),
+         y = seq(from = 1, to = length(wseq2), length = length(wseq2)),
+         z = xy, col = col, xlab = xlab, ylab = ylab,axes=FALSE, ...)
+         axis(1)
+
+          axis(4,at=length(wseq2)-axTicks(4)+1,labels=axTicks(4),las=2)
+          n <- length(wseq1)
+      text(1:n,n+1.5,wseq1,cex=0.75,xpd=TRUE,srt=90,adj=0)
+      m <- length(wseq2)
+      text(0,m:1,wseq2,cex=0.75,xpd=TRUE,adj=1)
+
+      if(label) {text(row(xy),col(xy),
+      wseq1,cex=0.75/wsize,col="white")
+      text(1:n,
+      m:1,
+      wseq2,col=2,cex=0.75/wsize)
+       }
+     box()
```

```
+     invisible(list(xy=xy,seq1=wseq1,seq2=wseq2))
+ }
```

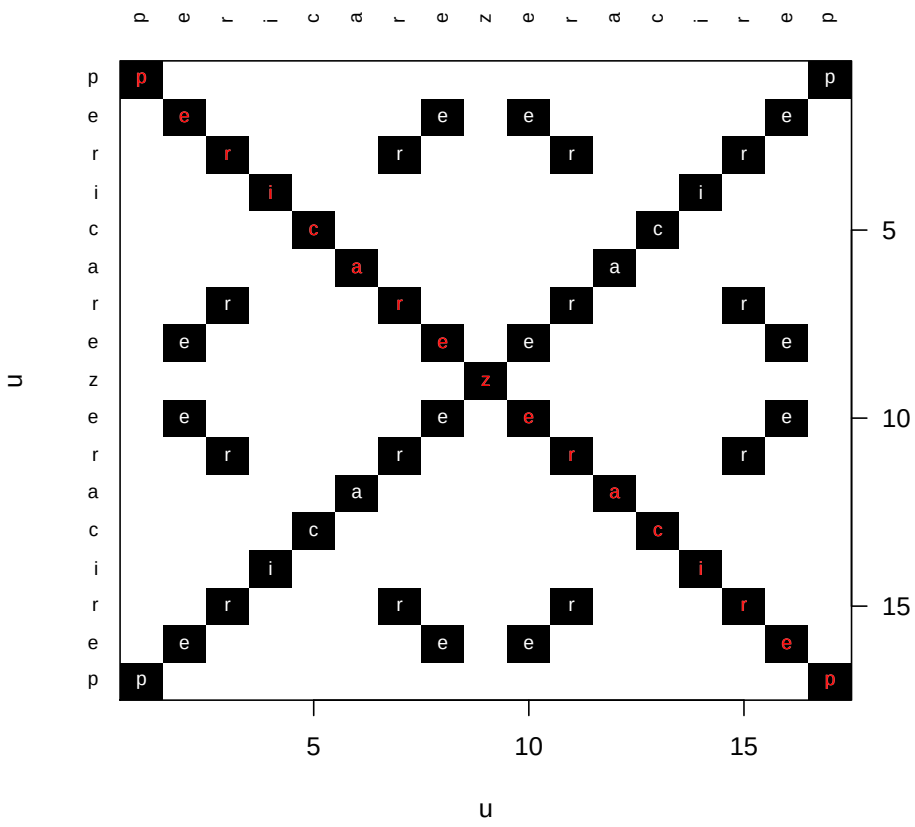
Prikaz s funkcijo `dotPlot()`

```
> dotPlot(v,v,wsize=1)
```



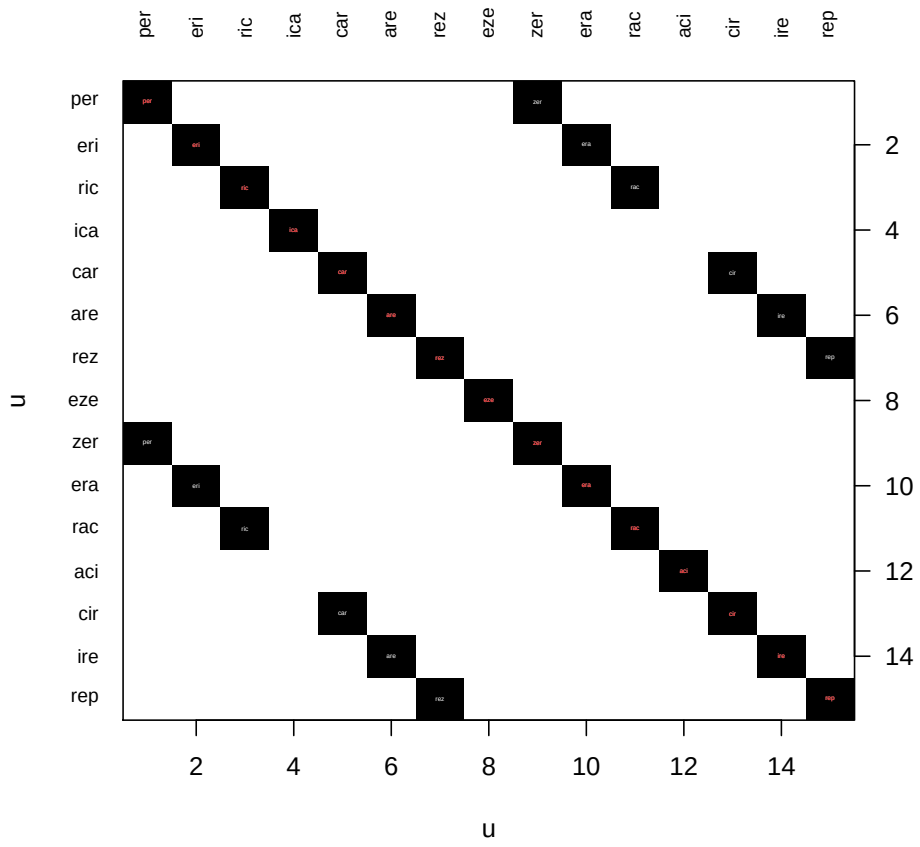
Palindrom

```
> seq2 <- "pericarezeracirep"
> (u<- s2c(seq2))
 [1] "p" "e" "r" "i" "c" "a" "r" "e" "z" "e" "r" "a" "c" "i" "r" "e"
[17] "p"
> dotPlot(u,u)
>
```



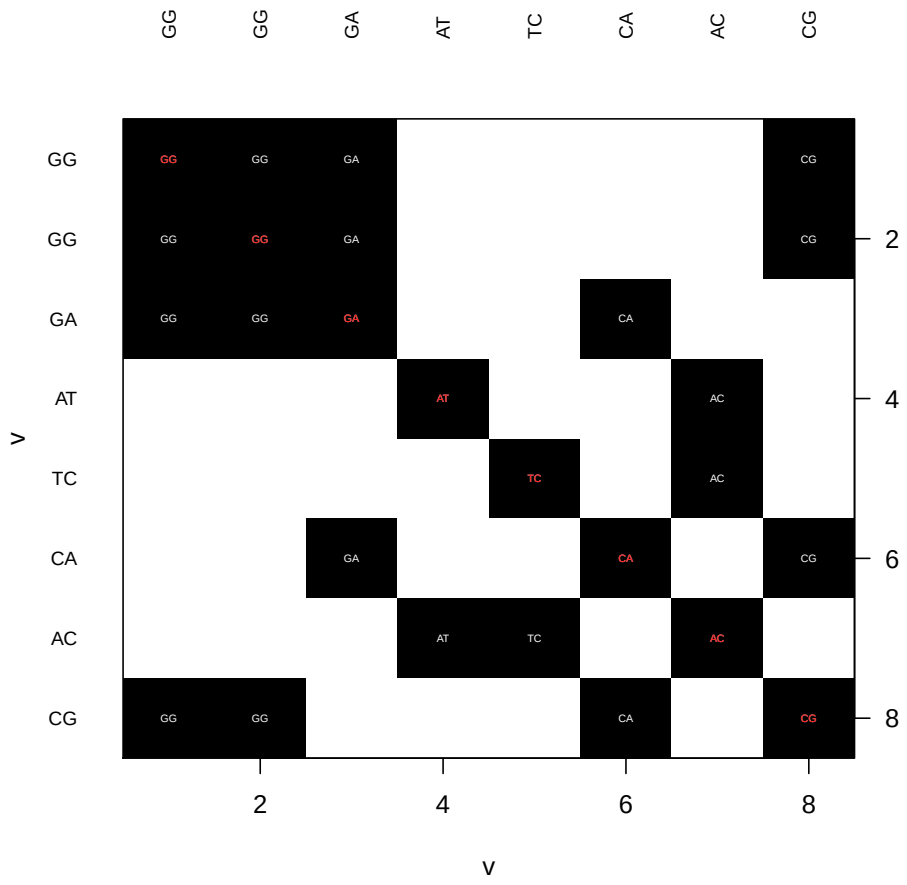
Primerjava z daljšim drsečim oknom

```
> dotPlot(u,u,wsize=3,nmatch=2)
>
```

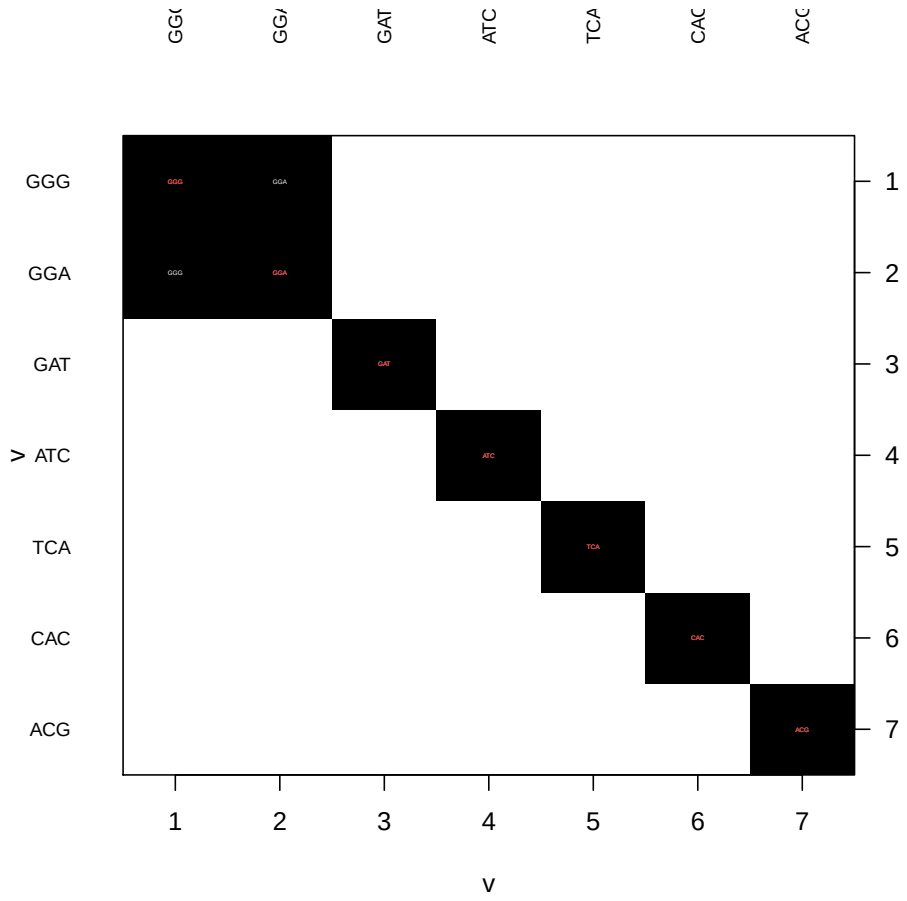


Daljše okno, sprememba števila ujemanj

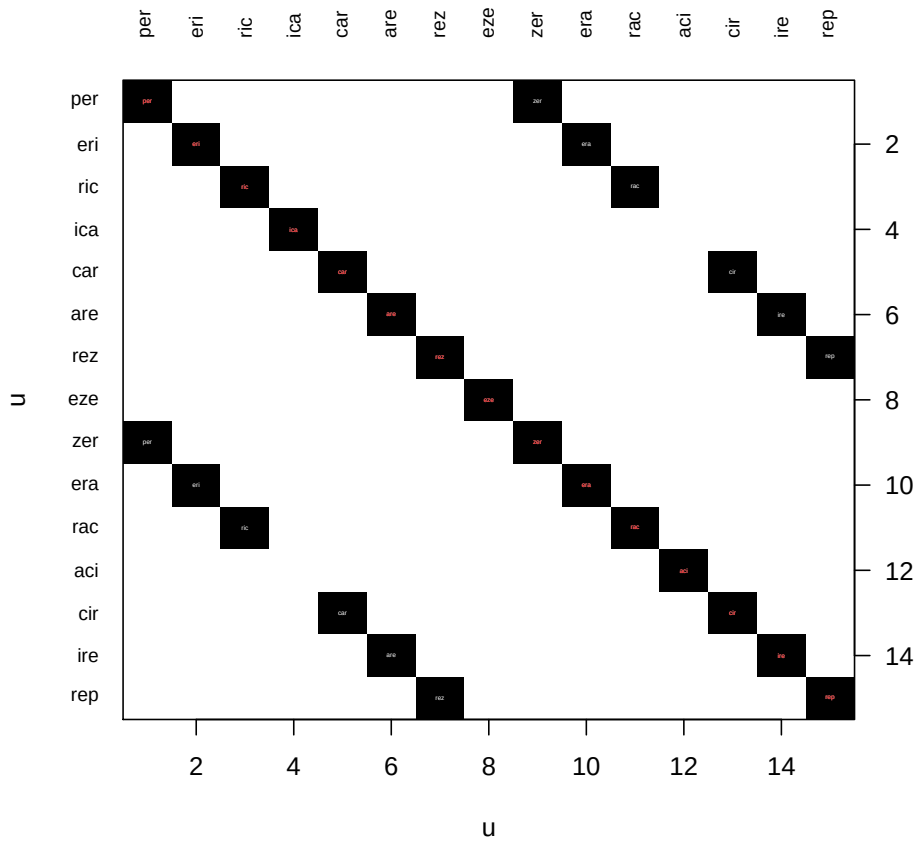
```
> dotPlot(v,v,wsize=2)
```




```
> dotPlot(v,v,wsize=3,nmatch=2)
```

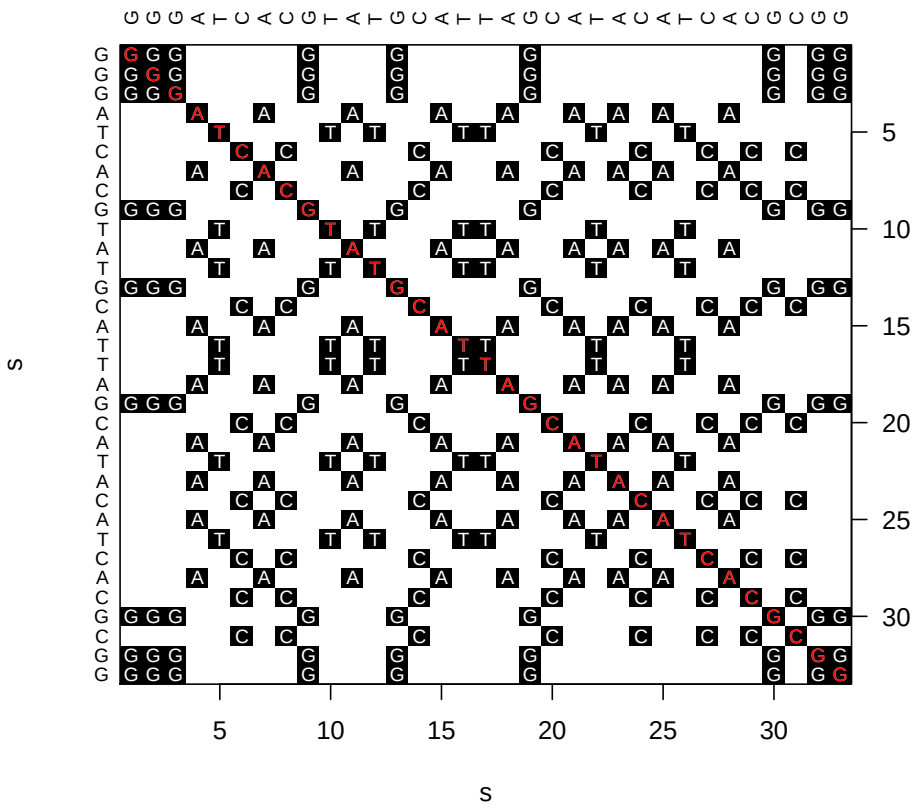


```
> seq2 <- "pericarezeracirep"
> (u<- s2c(seq2))
 [1] "p" "e" "r" "i" "c" "a" "r" "e" "z" "e" "r" "a" "c" "i" "r" "e"
[17] "p"
> dotPlot(u,u,wsize=3,nmatch=2)
>
```



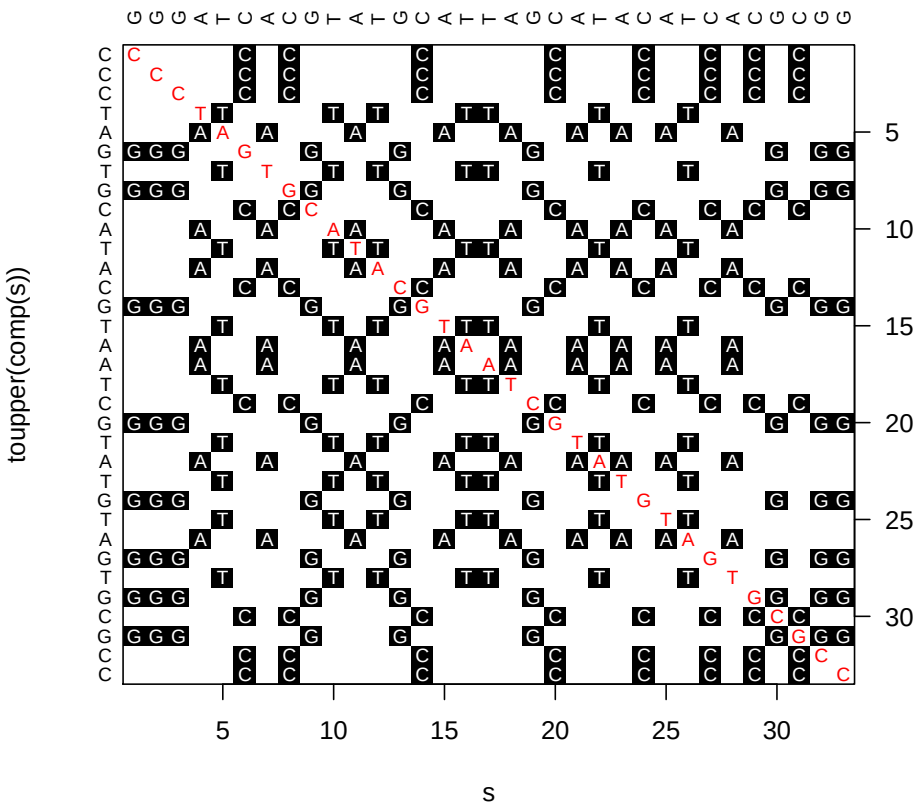
Daljša sekvenca

```
> seq1 <- "GGGATCACGTATGCATTAGCATACATCACGCGG"
> wsize <- 1
> (s <- s2c(seq1))
 [1] "G" "G" "G" "A" "T" "C" "A" "C" "G" "T" "A" "T" "G" "C" "A" "T"
[17] "T" "A" "G" "C" "A" "T" "A" "C" "A" "T" "C" "A" "C" "G" "C" "G"
[33] "G"
> dp <- dotPlot(s,s,wsize=wsize)
```



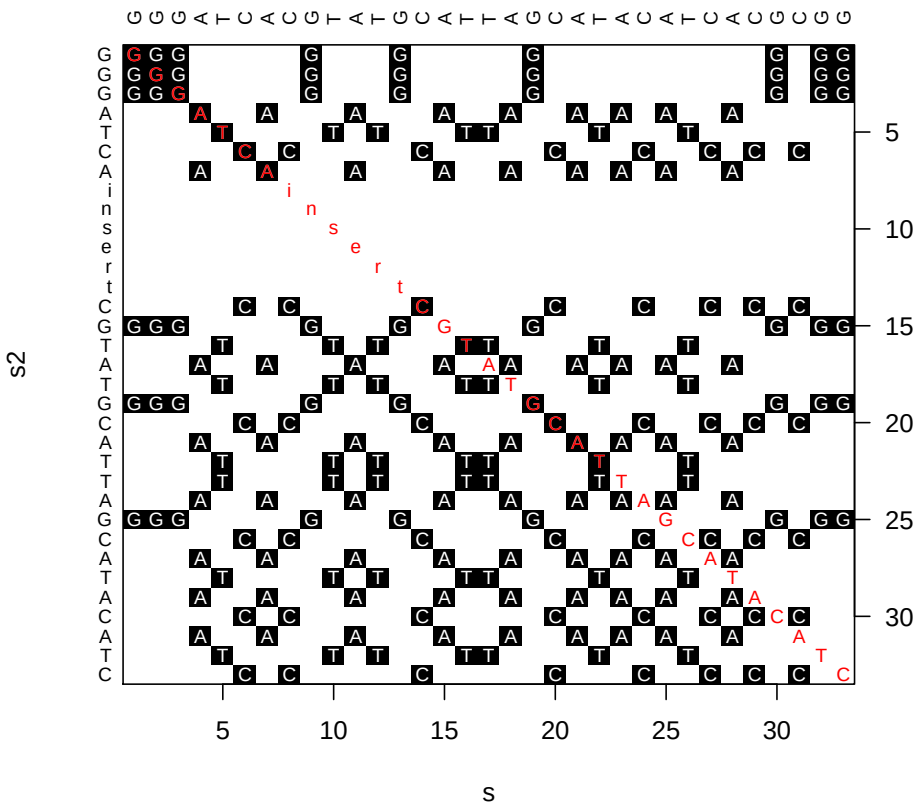
Funkcija `comp()` naredi komplement sekvence

```
> seq1 <- "GGGATCACGTATGCATTAGCATACATCACGCGG"
> wsize <- 1
> (s <- s2c(seq1))
 [1] "G" "G" "G" "A" "T" "C" "A" "C" "G" "T" "A" "T" "G" "C" "A" "T"
[17] "T" "A" "G" "C" "A" "T" "A" "C" "A" "T" "C" "A" "C" "G" "C" "G"
[33] "G"
> dp <- dotPlot(s,toupper(comp(s)),wsize=wsize)
```

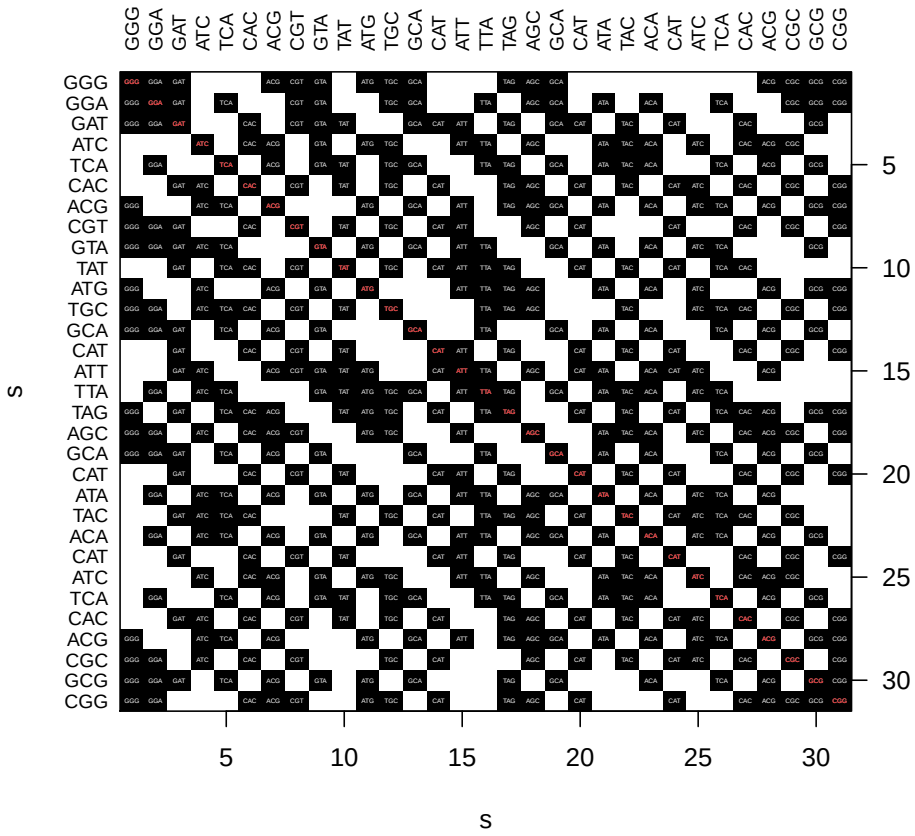


Vrinjena sekvenca

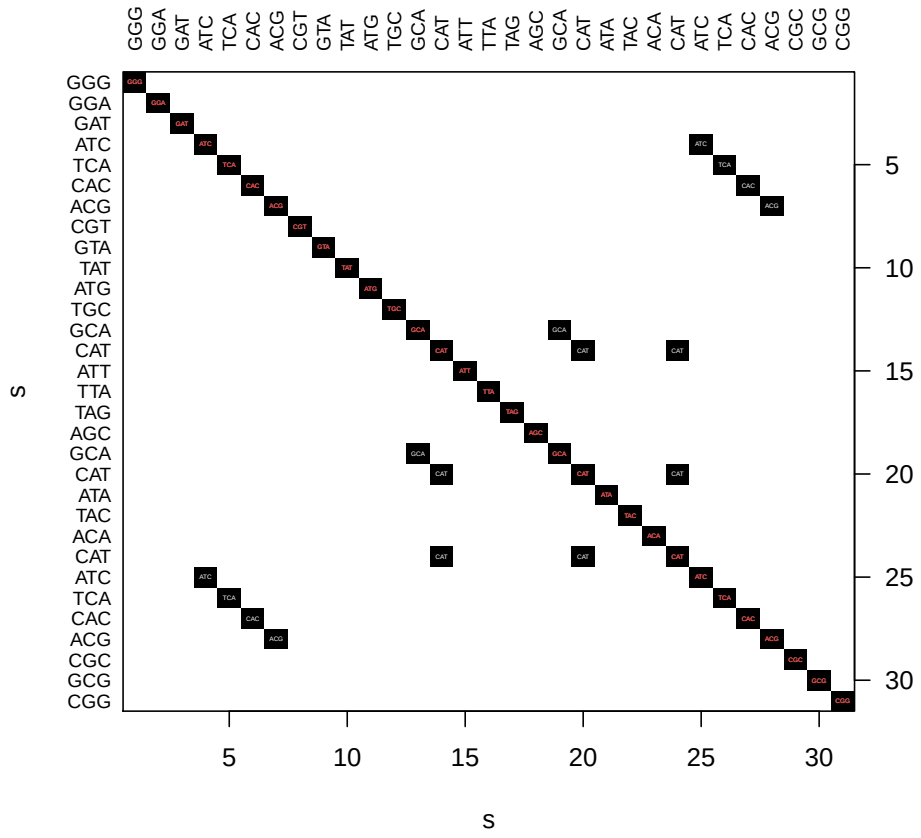
```
> seq1 <- "GGGATCACGTATGCATTAGCATACATCACGCGG"
> s <- s2c(seq1)
> s2 <-c(s[1:7],s2c("insert"),s[8:(length(s)-6)])
> seq1
[1] "GGGATCACGTATGCATTAGCATACATCACGCGG"
> c2s(s2)
[1] "GGGATCAinsertCGTATGCATTAGCATACATC"
> dotPlot(s,s2)
```




```
> seq1 <- "GGGATCACGTATGCATTAGCATACATCACGCGG"
> wsize <- 3
> (s <- s2c(seq1))
 [1] "G" "G" "G" "A" "T" "C" "A" "C" "G" "T" "A" "T" "G" "C" "A" "T"
[17] "T" "A" "G" "C" "A" "T" "A" "C" "A" "T" "C" "A" "C" "G" "C" "G"
[33] "G"
> dp <- dotPlot(s,s,wsize=wsize)
>
```




```
> seq1 <- "GGGATCACGTATGCATTAGCATACATCACGCGG"
> wsize <- 3
> nmatch <- 3
> (s <- s2c(seq1))
 [1] "G" "G" "G" "A" "T" "C" "A" "C" "G" "T" "A" "T" "G" "C" "A" "T"
[17] "T" "A" "G" "C" "A" "T" "A" "C" "A" "T" "C" "A" "C" "G" "C" "G"
[33] "G"
> dp <- dotPlot(s,s,wsize=wsize,nmatch=nmatch)
```


GGG GGA GAT ATC TCA CAC ACG CGT GTA TAT ATG TGC GCA CAT ATT TTA TAG AGC GCA CAT ATA TAC ACA CAT ATC TCA CAC ACG CGC GCG CGG
GGG GGA GAT ATC TCA CAC ACG CGT GTA TAT ATG TGC GCA CAT ATT TTA TAG AGC GCA CAT ATA TAC ACA CAT ATC TCA CAC ACG CGC GCG CGG
s
5 10 15 20 25 30
5 10 15 20 25 30
s

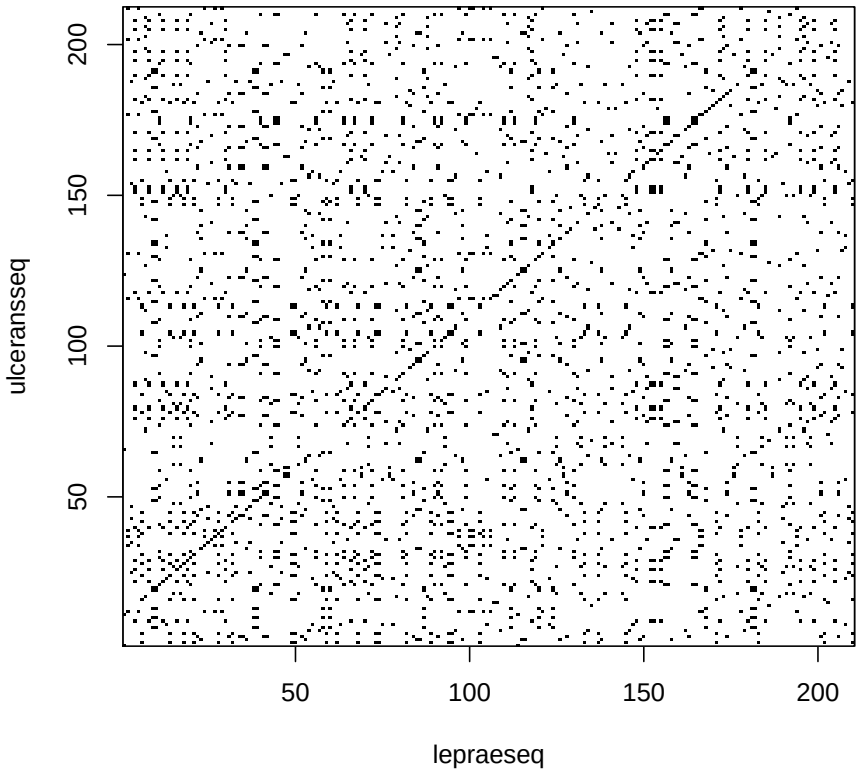
5 Privzem podatkov iz baze Swissprot

Izberemo dva proteina iz Swissprot. Accession Q9CD83 in A0PQ23.

```
> library("seqinr")
> choosebank("swissprot")
> query("leprae", "AC=Q9CD83")
> lepraeseq <- getSequence(leprae$req[[1]])
> query("ulcerans", "AC=A0PQ23")
> ulceransseq <- getSequence(ulcerans$req[[1]])
> closebank()
> lepraeseq # Display the contents of "lepraeseq"
  [1] "M" "T" "N" "R" "T" "L" "S" "R" "E" "E" "I" "R" "K" "L" "D" "R"
 [17] "D" "L" "R" "I" "L" "V" "A" "T" "N" "G" "T" "L" "T" "R" "V" "L"
 [33] "N" "V" "V" "A" "N" "E" "E" "I" "V" "V" "D" "I" "I" "N" "Q" "Q"
 [49] "L" "L" "D" "V" "A" "P" "K" "I" "P" "E" "L" "E" "N" "L" "K" "I"
 [65] "G" "R" "I" "L" "Q" "R" "D" "I" "L" "L" "K" "G" "Q" "K" "S" "G"
 [81] "I" "L" "F" "V" "A" "A" "E" "S" "L" "I" "V" "I" "D" "L" "L" "P"
 [97] "T" "A" "I" "T" "T" "Y" "L" "T" "K" "T" "H" "H" "P" "I" "G" "E"
[113] "I" "M" "A" "A" "S" "R" "I" "E" "T" "Y" "K" "E" "D" "A" "Q" "V"
[129] "W" "I" "G" "D" "L" "P" "C" "W" "L" "A" "D" "Y" "G" "Y" "W" "D"
[145] "L" "P" "K" "R" "A" "V" "G" "R" "R" "Y" "R" "I" "I" "A" "G" "G"
[161] "Q" "P" "V" "I" "I" "T" "T" "E" "Y" "F" "L" "R" "S" "V" "F" "Q"
[177] "D" "T" "P" "R" "E" "E" "L" "D" "R" "C" "Q" "Y" "S" "N" "D" "I"
[193] "D" "T" "R" "S" "G" "D" "R" "F" "V" "L" "H" "G" "R" "V" "F" "K"
[209] "N" "L"
```

Dotplot - diagonala kaže enake aminokisline na podobnih ali enakih mestih v obeh proteinih.

```
> rm(dotPlot) # odstranim svoj dotPlot
> dotPlot(lepraeseq, ulceransseq)
```



6 Bioconductor in paket Biostrings

Za poravnave sekvenc lahko uporabite paket **Biostrings**, ki je del obširnega in dobro dokumentiranega sistema *Bioconductor*.

6.1 Paket Biostrings

Nekaj o tem si lahko preberete v poglavju Pairwise alignement v spletni knjigi Little Book of R for Bioinformatics (Coghlan, 2012).

6.2 Poravnanve DNA zaporedij z Needleman-Wunsch algoritmom

```
> library(Biostrings)
> sigma <- nucleotideSubstitutionMatrix(
+           match = 2, mismatch = -1, baseOnly = TRUE)
> sigma # Print out the matrix
```

```
   A  C  G  T
A  2 -1 -1 -1
C -1  2 -1 -1
G -1 -1  2 -1
T -1 -1 -1  2
```

Optimalna poravnava

```
> s1 <- "GAATTC"
> s2 <- "GATTA"
> globalAligns1s2 <- pairwiseAlignment(
+ s1, s2, substitutionMatrix = sigma, gapOpening = -2,
+ gapExtension = -8, scoreOnly = FALSE)
> globalAligns1s2 # Print out the optimal alignment and its score
Global PairwiseAlignedFixedSubject (1 of 1)
pattern: [1] GAATTC
subject: [1] GA-TTA
score: -3
```

6.3 Poravnanve zaporedij proteinov z Needleman-Wunsch algoritmom

```
>  data(BLOSUM50)
>  BLOSUM50 # Print out the data
   A  R  N  D  C  Q  E  G  H  I  L  K  M  F  P  S  T  W  Y  V  B  Z
A  5 -2 -1 -2 -1 -1 -1  0 -2 -1 -2 -1 -1 -3 -1  1  0 -3 -2  0 -2 -1
R -2  7 -1 -2 -4  1  0 -3  0 -4 -3  3 -2 -3 -3 -1 -1 -3 -1 -3 -1  0
N -1 -1  7  2 -2  0  0  0  1 -3 -4  0 -2 -4 -2  1  0 -4 -2 -3  4  0
D -2 -2  2  8 -4  0  2 -1 -1 -4 -4 -1 -4 -5 -1  0 -1 -5 -3 -4  5  1
C -1 -4 -2 -4 13 -3 -3 -3 -3 -2 -2 -3 -2 -2 -4 -1 -1 -5 -3 -1 -3 -3
Q -1  1  0  0 -3  7  2 -2  1 -3 -2  2  0 -4 -1  0 -1 -1 -1 -3  0  4
E -1  0  0  2 -3  2  6 -3  0 -4 -3  1 -2 -3 -1 -1 -1 -3 -2 -3  1  5
G  0 -3  0 -1 -3 -2 -3  8 -2 -4 -4 -2 -3 -4 -2  0 -2 -3 -3 -4 -1 -2
H -2  0  1 -1 -3  1  0 -2 10 -4 -3  0 -1 -1 -2 -1 -2 -3  2 -4  0  0
I -1 -4 -3 -4 -2 -3 -4 -4 -4  5  2 -3  2  0 -3 -3 -1 -3 -1  4 -4 -3
L -2 -3 -4 -4 -2 -2 -3 -4 -3  2  5 -3  3  1 -4 -3 -1 -2 -1  1 -4 -3
K -1  3  0 -1 -3  2  1 -2  0 -3 -3  6 -2 -4 -1  0 -1 -3 -2 -3  0  1
M -1 -2 -2 -4 -2  0 -2 -3 -1  2  3 -2  7  0 -3 -2 -1 -1  0  1 -3 -1
F -3 -3 -4 -5 -2 -4 -3 -4 -1  0  1 -4  0  8 -4 -3 -2  1  4 -1 -4 -4
P -1 -3 -2 -1 -4 -1 -1 -2 -2 -3 -4 -1 -3 -4 10 -1 -1 -4 -3 -3 -2 -1
S  1 -1  1  0 -1  0 -1  0 -1 -3 -3  0 -2 -3 -1  5  2 -4 -2 -2  0  0
T  0 -1  0 -1 -1 -1 -1 -2 -2 -1 -1 -1 -1 -2 -1  2  5 -3 -2  0  0 -1
W -3 -3 -4 -5 -5 -1 -3 -3 -3 -3 -2 -3 -1  1 -4 -4 -3 15  2 -3 -5 -2
Y -2 -1 -2 -3 -3 -1 -2 -3  2 -1 -1 -2  0  4 -3 -2 -2  2  8 -1 -3 -2
V  0 -3 -3 -4 -1 -3 -3 -4 -4  4  1 -3  1 -1 -3 -2  0 -3 -1  5 -4 -3
B -2 -1  4  5 -3  0  1 -1  0 -4 -4  0 -3 -4 -2  0  0 -5 -3 -4  5  2
Z -1  0  0  1 -3  4  5 -2  0 -3 -3  1 -1 -4 -1  0 -1 -2 -2 -3  2  5
X -1 -1 -1 -1 -2 -1 -1 -2 -1 -1 -1 -1 -1 -2 -2 -1  0 -3 -1 -1 -1 -1
* -5 -5 -5 -5 -5 -5 -5 -5 -5 -5 -5 -5 -5 -5 -5 -5 -5 -5 -5 -5 -5 -5
   X  *
A -1 -5
R -1 -5
N -1 -5
D -1 -5
C -2 -5
Q -1 -5
E -1 -5
G -2 -5
H -1 -5
I -1 -5
```

```
L -1 -5
K -1 -5
M -1 -5
F -2 -5
P -2 -5
S -1 -5
T  0 -5
W -3 -5
Y -1 -5
V -1 -5
B -1 -5
Z -1 -5
X -1 -5
* -5  1
```

Matrike vrednotenja zamenjav

```
> data(package="Biostrings")
```

```
>  data(BLOSUM50)
>  s3 <- "PAWHEAE"
>  s4 <- "HEAGAWGHEE"
>  globalAligns3s4 <- pairwiseAlignment(
+  s3, s4, substitutionMatrix = "BLOSUM50", gapOpening = -2,
+ gapExtension = -8, scoreOnly = FALSE)
>  globalAligns3s4 # Print out the optimal global alignment and its score
Global PairwiseAlignedFixedSubject (1 of 1)
pattern: [1] P---AWHEAE
subject: [1] HEAGAWGHEE
score: -5
```

6.4 Daljša poravnava

Pretvorba v nize znakov

```
> lepraeseqstring <- c2s(lepraeseq)      # Make a string that contains the sequ
> ulceransseqstring <- c2s(ulceransseq) # Make a string that contains the sequ
> ulceransseq[1:15]
 [1] "M" "L" "A" "V" "L" "P" "E" "K" "R" "E" "M" "T" "E" "C" "H"
> ulceransseqstring
[1] "MLAVLPEKREMTECHLSDEEIRKLNRDLRILIATNGTLTRILNVLANDEIVVEIVKQQIQDAAPEMDGCDHSS
```

Če je potrebno, spremenimo v velike črke

```
> lepraeseqstring <- toupper(lepraeseqstring)
> ulceransseqstring <- toupper(ulceransseqstring)

> c(lepraeseqstring,ulceransseqstring)
[1] "MTNRTLSREEIRKLDRDLRILVATNGTLTRVLNVVANEEIVVDIINQQLLDVAPKIPELENLKIGRILQRDIL
[2] "MLAVLPEKREMTECHLSDEEIRKLNRDLRILIATNGTLTRILNVLANDEIVVEIVKQQIQDAAPEMDGCDHSS
```

Poravnava

```
> globalAlignLepraeUlcerans <- pairwiseAlignment(
+ lepraeseqstring, ulceransseqstring,
+ substitutionMatrix = BLOSUM50,
+ gapOpening = -2, gapExtension = -8, scoreOnly = FALSE)
> globalAlignLepraeUlcerans # Print out the optimal global alignment and its s
Global PairwiseAlignedFixedSubject (1 of 1)
pattern: [1] MT-----NR--T---LSREEIRKLDRDLRI...EELDRCQYSNDIDTRSGDRFVLHGRVFKN
subject: [1] MLAVLPEKREMTECHLSDEEIRKLNRDLRI...EEPIRHQRS--VGT-SA-R---SGRSICT
score: 627
```

References

Charif, D. and J. Lobry (2007). SeqinR 1.0-2: a contributed package to the R project for statistical computing devoted to biological sequences retrieval and analysis. In H. R. U. Bastolla, M. Porto and M. Vendruscolo (Eds.), *Structural approaches to sequence evolution: Molecules, networks, populations*, Biological and Medical Physics, Biomedical Engineering, pp. 207–232. New York: Springer Verlag. ISBN : 978-3-540-35305-8. 1

Coghlan, A. (2012). Little book of R for bioinformatics. http://a-little-book-of-r-for-bioinformatics.readthedocs.org/en/latest/. 23

SessionInfo

Windows 7 x64 (build 7601) Service Pack 1

- R version 2.15.1 (2012-06-22), `x86_64-pc-mingw32`
- Locale: `LC_COLLATE=Slovenian_Slovenia.1250`, `LC_CTYPE=Slovenian_Slovenia.1250`, `LC_MONETARY=Slovenian_Slovenia.1250`, `LC_NUMERIC=C`, `LC_TIME=Slovenian_Slovenia.1250`
- Base packages: base, datasets, graphics, grDevices, methods, splines, stats, utils
- Other packages: Biostrings 2.22.0, Hmisc 3.9-3, IRanges 1.12.6, patchDVI 1.8.1584, seqinr 3.0-6, survival 2.36-14
- Loaded via a namespace (and not attached): cluster 1.14.2, grid 2.15.1, lattice 0.20-6, tools 2.15.1

Project path: `D:/_Y/R/Bioinformatika`

View as vignette

Project files can be viewed by pasting this code to R console:

```
> projectName <-"Bioinformatika";  mainFile <-"dotplot"

> commandArgs()
> library(tkWidgets)
> # getrootpath <- function() {
> # fp <- (strsplit(getwd(), "/"))[[1]]
> # file <- file.path(paste(fp[-length(fp)], collapse = "/"))
> # return(file)
> # }
> # fileName <- function(name="bla",ext="PDF") paste(name,ext,sep=".")
>  openPDF(file.path(dirname(getwd()),"doc",paste(mainFile,"PDF",sep=".")))
>  viewVignette("viewVignette", projectName, file.path("../doc",paste(mainFile
>
```