

# Manipulacija nizov znakov

A. Blejec

27. oktober 2009

## Iskanje lege baz

```
> baze <- c("A", "T", "C", "G")
> n <- 100
> dna <- sample(baze, n, replace = TRUE)
> dna
 [1] "A" "C" "C" "C" "G" "C" "A" "A" "C" "C" "C"
[12] "C" "T" "G" "T" "G" "T" "T" "A" "A" "T" "T"
[23] "A" "A" "A" "G" "C" "G" "G" "A" "T" "T" "T"
[34] "C" "A" "G" "A" "T" "G" "G" "C" "C" "T" "C"
[45] "T" "C" "C" "T" "A" "G" "A" "T" "C" "C" "A"
[56] "C" "T" "G" "A" "G" "G" "A" "T" "A" "A" "C"
[67] "T" "C" "A" "C" "A" "G" "A" "G" "A" "C" "T"
[78] "A" "T" "C" "G" "T" "A" "C" "A" "G" "T" "T"
[89] "A" "G" "A" "G" "A" "A" "A" "C" "T" "A" "T"
[100] "C"
```

```
> grep("A", dna)
```

```
 [1]  1  7  8 19 20 23 24 25 30 35 37 49 51 55 59
[16] 62 64 65 69 71 73 75 78 83 85 89 91 93 94 95
```

## Kje se pojavljajo baze

```
> sapply(baze, grep, dna)
```

```
$A
```

```
[1] 1 7 8 19 20 23 24 25 30 35 37 49 51 55 59  
[16] 62 64 65 69 71 73 75 78 83 85 89 91 93 94 95  
[31] 98
```

```
$T
```

```
[1] 13 15 17 18 21 22 31 32 33 38 43 45 48 52 57  
[16] 63 67 77 79 82 87 88 97 99
```

```
$C
```

```
[1] 2 3 4 6 9 10 11 12 27 34 41  
[12] 42 44 46 47 53 54 56 66 68 70 76  
[23] 80 84 96 100
```

```
$G
```

```
[1] 5 14 16 26 28 29 36 39 40 50 58 60 61 72 74  
[16] 81 86 88 89
```

## Kje se pojavlja triplet nukleotidov?

```
> dna <- paste(dna, collapse = "")
> ind <- gregexpr("AC[TC]", dna)[[1]]
> ind
[1] 1 8 55 65 75 95
attr(,"match.length")
[1] 3 3 3 3 3 3
> for (i in ind) cat(i, " ", substr(dna,
+   i, i + 2), "\n")
1 ACC
8 ACC
55 ACT
65 ACT
75 ACT
95 ACT
> dna
[1] "ACCCGCAACCCCTGTGTTAATTAAAGCGGATTTTCAGATGGCCTCTCCTAGA
```

# Komplementarna kopija

Zamenjam z ustrežno malo črko

```
> dna
```

```
[1] "ACCCGCAACCCCTGTGTTAATTAAAGCGGATTCAGATGGCCTCTCCTAGA"
```

```
> baze
```

```
[1] "A" "T" "C" "G"
```

```
> obrat <- c("t", "a", "g", "c")
```

```
> names(obrat) <- baze
```

```
> obrat
```

```
  A   T   C   G  
"t" "a" "g" "c"
```

```
> gsub("A", obrat["A"], dna)
```

```
[1] "tCCCGCttCCCCTGTGTTttTTtttGCGGtTTTctGtTGGCCTCTCCTtGt"
```

## Komplement vseh baz

```
> dna
[1] "ACCCGCAACCCCTGTGTTAATTAAAGCGGATTTTCAGATGGCCTCTCCTAGA
> obrat <- c("t", "a", "g", "c")
> names(obrat) <- baze
> obrat
  A   T   C   G
"t" "a" "g" "c"
> reverse <- dna
> for (znak in baze) reverse <- gsub(znak,
+   obrat[znak], reverse)
> reverse
[1] "tgggcgttggggacacaattaatttcgcctaaagtctaccggagaggatct
> reverse <- toupper(reverse)
> cat("", reverse, "\n", dna, "\n")
TGGGCGTTGGGGACACAATTAATTTTCGCCTAAAGTCTACCGGAGAGGATCTAGGT
ACCCGCAACCCCTGTGTTAATTAAAGCGGATTTTCAGATGGCCTCTCCTAGATCCA
```

## Funkcija za komplementarno verigo

### Definicija funkcije

```
> complement <- function(dna) {  
+   baze <- c("A", "T", "C", "G")  
+   obrat <- c("t", "a", "g", "c")  
+   names(obrat) <- baze  
+   reverse <- dna  
+   for (znak in baze) {  
+     reverse <- gsub(znak, obrat[znak],  
+       reverse)  
+   }  
+   reverse <- toupper(reverse)  
+   return(reverse)  
+ }
```

### Uporaba

```
> complement(dna)
```

```
[1] "TGGGCGTTGGGGACACAATTAATTTTCGCCTAAAGTCTACCGGAGAGGATCT"
```

## Pretvorba niza v vektor znakov

### toChar

```
> toChar <- function(x, first = 1:nchar(x),  
+   last = 1:nchar(x)) {  
+   x <- substring(x, first = first, last = last)  
+   return(x)  
+ }
```

### Example

```
> toChar("ACTTCACG")  
[1] "A" "C" "T" "T" "C" "A" "C" "G"  
  
> sapply(c("ACT", "TCAG"), toChar)  
  
$ACT  
[1] "A" "C" "T"  
  
$TCAG  
[1] "T" "C" "A" "G"
```