

# KEGG anotacija

A. Blejec

November 4, 2009

## Contents

### 1 Dostop do KEGG Kyoto Encyclopedia of Genes and Genomes

Dostop do [KEGG](#) v R omogoča paket **KEGGSOAP**, ki ga najdemo na [Biocunductor](#) strani. Povezava do KEGG temelji na protokolu SOAP (Simple Object Access Protocol).

S pomočjo funkcije `bget` lahko dosežemo dodatne informacije, ki jih prilepimo k opisu genov. Zaradi `Pathways` je zanimiv `NCBI-GeneID`, ki ga lahko povežemo s podpoljem `Locus` v polju `Name` GAL datoteke.

```
> eset <- allData[1:5, ]
> gName <- featureData(eset)$Name
> gName

[1] Locus:b2587|Gene:kgtP|Prot:NP_417082_428_462
[2] Locus:b1947|Gene:fliO|Prot:NP_416457_2_36
[3] Locus:b0454|Gene:ybaZ|Prot:NP_414988_257_291
[4] Locus:b1603|Gene:pntA|Prot:NP_416120_790_824
[5] Locus:b0368|Gene:tauD|Prot:NP_414902_609_643
12041 Levels: <no oligo> ... QC-oligoB-AS-3

> grep("^Locus:", gName)

[1] 1 2 3 4 5

> Locus <- sub("^Locus:(b[0-9]*)[:print:]*", "\\1", gName)
> Locus[1:5]

[1] "b2587" "b1947" "b0454" "b1603" "b0368"
```

## 1.1 Dinamičen prevod posameznih kod

Takole bi lahko dinamično poizvedeli kakšne so posamezne kode za NCBI-GeneID:

```
> library(KEGGSOAP)
> n <- 5
> GeneID <- character(n)
> system.time(for (i in 1:n) {
+   x <- bget(paste("eco", Locus[i], sep = ":"))
+   if (length(x) > 0) {
+     first <- substring.location(x, "NCBI-GeneID: ")$last +
+     1
+     last <- substring.location(x, "\n", restrict = c(first,
+     9999))$first[1] - 1
+     GeneID[i] <- substring(x, first, last)
+   }
+ })
   user  system elapsed
   0.52   0.04   7.39
> cbind(Locus, GeneID)
   Locus  GeneID
[1,] "b2587" "947069"
[2,] "b1947" "946458"
[3,] "b0454" "945094"
[4,] "b1603" "946628"
[5,] "b0368" "945021"
```

Za eno poizvedbo rabi malo več kot 1 sekundo.

## 1.2 Poizvedba za več genov naenkrat

Z eno poizvedbo lahko dobimo informacijo o več genih naenkrat, (nekje piše da največ 100). Za neobstoječe KEGG kode vrne prazen niz.

```
> eset <- allData[featureData(allData)$Status == "gene",
+ ]
> M <- min(100, dim(eset)[1])
> eset <- eset[1:M, ]
> gName <- featureData(eset)$Name
> Locus <- sub("^Locus: (b[0-9]*) [[:print:]]*", "\\1", gName)
> Locus[1:5]
[1] "b2587" "b1947" "b0454" "b1603" "b0368"
> stime <- system.time(seznam <- strsplit(bget(paste("eco",
+   Locus, sep = ":", collapse = " ")), "ENTRY"))
> stime
   user  system elapsed
   0.11   0.02   5.82
```

Dobili smo seznam kod v obliki list, preuredimo jo v matriko, pri čemer izpustimo redundantno vrstico.

```
> str(seznam)
```

List of 1

```
§ : chr [1:92] "" " b2587 CDS E.coli\nNAME kgt  
> s <- as.matrix(seznam[[1]][-1], ncol = 1)
```

Za 92 kod je potreboval 5.82 sekund, tako da bi za vseh 12240 genov potreboval približno 13 minut. Učinkoviteje, ampak še vedno veliko!

Vrne eno dodatno prazno vrstico na začetku. Pri nas imamo na mestu 95 neobstoječo kodo b2999, tako da je vrstni red rezultata drugačen od vrstnega reda vhodnih kod!!! Za neobstoječe ali neveljavne kode vrne prazen niz, zato vrstice ustrezajo zaporedju veljavnih KEGG kod. Lahko izločimo neveljavne (pri nas tovarniške kontrole) ali pa naredim preslikavo na podatke s pomočjo KEGG kode.

S funkcijo bget dobimo vso informacijo iz KEGG baze!

```
> cat (seznam[[1]][2])
```

```
      b2587          CDS      E.coli  
NAME      kgtP, ECK2585, JW2571, witA  
DEFINITION alpha-ketoglutarate transporter  
ORTHOLOGY KO: K03761 MFS transporter, MHS family, alpha-ketoglutarate  
           permease  
POSITION  complement (2722470..2723768)  
MOTIF     Pfam: CitMHS Sugar_tr MFS_1 LacY_symp DUF2298 DUF1345 BPD_transp_1  
           PUCG Voltage_CLC 7TM_GPCR_Srh Branch_AA_trans DUF112  
           Nucleos_tra2_C Flavi_NS2A Frag1 Exosortase_EpSH DUF340 DUF92  
           DUF205 Competence DUF125 Ferric_reduct  
PROSITE:  SUGAR_TRANSPORT_1 SUGAR_TRANSPORT_2 MFS  
DBLINKS   RegulonDB: B2587  
           EcoGene: EG10522  
           ECOCYC: EG10522  
           NCBI-GI: 16130512  
           NCBI-GeneID: 947069  
           UniProt: P0AEX3  
CODON_USAGE      T          C          A          G  
T   13   16   13   8   4   3   8   10   12   4   0   1   6   1   0   9  
C   4    7    6  18   4   0   3   2   3   2   8   5   8  12   0   0  
A   9    5    7  19   9   8   4   6   4   3   9   4   7   2   1   0  
G  14    4    4  12  14  14  11  12   5   2  10   4  14   9  12   5  
AASEQ           432  
MAESTVTADSKLTSSDTRRRIWAIIVGASSGNLVEWFDYFVYSFCSLYFAHIFFPNGNTTT  
QLLQTAGVFAAGFLMRPIGGWLFGRADKHGRKKSMLLSVCMCFGSLVIACLPGYETIG  
TWAPALLLLARLFQGLSVGGEYGTSAATMSEVAVEGRKGFYASFQYVTLIGGQLLALLV  
VVLQHTMEDAALREWGWRIPFALGAVLAVVALWLRRLDQDETSQOETRALKEAGSLKGLWR  
NRRAFIMVLGFTAAGSLCFYFTTMYQKYLVTAGMHANVASGIMTAALFVFMLIQPLIG  
ALSDKIGRRTSMLCFGSLAAIFTVPILSALQNVSSPYAAFGLVMCALLIVSFYTSISGIL  
KAEMFPAQVRALGVGLSYAVANAIFGGSAEYVALSLKSIGMETAFFWYVTLMAVVAFLVS  
LMLHRKKGKGMRL  
NTSEQ           1299  
atggctgaaagtactgtaacggcagacagcaactgacaagtagtgatactcgctcgccgc  
atctgggcgattgtggggcctcttcaggtaactctggtcgagtggttcgattctatgct  
tactcgttctgttcaactctactttgccacatcttcttccctccgggaacacgacgact  
caactactacaaacagcaggtgtttttgctgcgggattcctgatgcccgaataggcgt  
tggctatctggccgatagccgataaacatggtcgcaaaaatcgatgctggtatcggtg  
tgtatgatgtgttccgatcgctggttatcgctgcctcccagggttatgaaactatagg  
acgtgggctccggcattattgcttctcgctcgtttatttcagggttatctgttggcgga
```

```

gaatatggcaccagcgccacctatatgagtgaagttgccggtgaagggcgcaaaggttt
tacgcatcatttcagtatgtgacggtgatcgccggacaactgctagccctactggttgc
gtggtttacaacacaccatggaagacgctgactcagagagtggggatggcgtattcct
ttcgcgtaggagctgtgtagctggtgtggcggtgtggttacgtcgtcagttagatgaa
acttcgcaacaagaaacgcgcgctttaaagaagctggatctctgaaaggattatggcgc
aatcgccgtgcattcatcatggttctcggtttaccgctgcgggctccctttgtttctat
accttcactacttatatgcagaagtatctggtaaatactgcgggaatgcatgccaacgtg
gcgagtggcattatgactgccgcattggttgtattcatgcttattcaaccactcattggc
gcgctgtcggataagattggtcgccgtacctcaatggtatggttcggttcgctggcagcc
atthttaccgttcctattctctcagcattgcaaaacgttcctcgccttatgccgctttt
ggtctggtgatgtgtgccctgctgatagtgagttttatacatcaatcagtggaatactg
aaggctgagatggtcccggcacaggttcgcgcattaggcgttggtctgtcatatgcggtc
gctaatgctatatttggtggttcggcggagtagtagcgttgctcgtgaaatcaatagga
atggaaacagccttcttctggtatgtgacctgatggccgtgggtggcgtttctggtttct
ttgatgctacatcgcaaaggggaaggggatgcgtcttag

```

```
///
```

### 1.3 Izbira kod iz seznama: funkcija getCode

S funkcijo getCode lahko iz seznama dobimo NCBI-GeneID kode:

```

> getCode <- function(x, str1 = "NCBI-GeneID: ", str2 = "\n") {
+   first <- function(x, str = "NCBI-GeneID: ") substring.location(x,
+     str)$last + 1
+   last <- function(x, str = "\n", start = 1) (substring.location(x,
+     str, restrict = c(start, 9999))$first)[1] - 1
+   st <- first(x, str1)
+   en <- last(x, str2, start = st)
+   return(substring(x, st, en)[1])
+ }

```

Ob klicu funkcije getCode definiramo enoličen niz znakov pred kodo in niz, ki kodo zaključuje.

```

> GeneID <- apply(s, 1, getCode, "NCBI-GeneID: ", "\n")
> GeneID[1:5]
[1] "947069" "946458" "945094" "946628" "945021"

```

ali pa KEGG kode:

```

> KEGGId <- paste("b", apply(s, 1, getCode, "b", " "),
+   sep = "")
> KEGGId[1:5]
[1] "b2587" "b1947" "b0454" "b1603" "b0368"

```

in ECOCYC kode:

```

> ecocycId <- apply(s, 1, getCode, "ECOCYC: ")
> ecocycId[1:5]
[1] "EG10522" "EG11224" "G6251" "EG10744" "EG12423"

```

Združena tabela

```
> print(cbind(KEGGId, GeneID, ecocycId)[1:5, ], quote = FALSE)
```

```

      KEGGId GeneID ecocycId
[1,] b2587  947069 EG10522
[2,] b1947  946458 EG11224
[3,] b0454  945094 G6251
[4,] b1603  946628 EG10744
[5,] b0368  945021 EG12423

```

Funkcijo `getCode` lahko uporabimo tudi na sestavljenih imenih genov:

```

> gName <- as.matrix(featureData(eset)$Name[1:5], ncol = 1)
> gName

      [,1]
[1,] "Locus:b2587|Gene:kgtP|Prot:NP_417082_428_462"
[2,] "Locus:b1947|Gene:fliO|Prot:NP_416457_2_36"
[3,] "Locus:b0454|Gene:ybaZ|Prot:NP_414988_257_291"
[4,] "Locus:b1603|Gene:pntA|Prot:NP_416120_790_824"
[5,] "Locus:b0368|Gene:tauD|Prot:NP_414902_609_643"

```

S primernimi oznakami začetkov in koncev lahko izluščimo kode:

```

> apply(gName, 1, getCode, "Locus:", "|")
[1] "b2587" "b1947" "b0454" "b1603" "b0368"

> apply(gName, 1, getCode, "Gene:", "|")
[1] "kgtP" "fliO" "ybaZ" "pntA" "tauD"

```

Včasih je treba biti nekoliko zvit :)

```

> paste("NP", apply(gName, 1, getCode, "Prot:NP_", "_"),
+       sep = "_")
[1] "NP_417082" "NP_416457" "NP_414988" "NP_416120" "NP_414902"

```

## 1.4 Izbira imenovanih kod z regularnimi izrazi (regular expression)

S pomočjo regularnih izrazov in funkcijo `sub` lahko učinkovito izluščimo kode. Pripravimo seznam lokusov

```
> gName <- featureData(eset)$Name
> Locus <- sub("^Locus: (b[0-9]*) .*", "\\1", gName)
> tail(Locus)
[1] "b3862" "b1275" "b2791" "b1942" "b1343" "b4558"
```

Privzemimo opise s spletne strani KEGG.

```
> system.time(KEGGstr <- bget(paste("eco", Locus, sep = ":",
+   collapse = " ")))
   user  system elapsed
  0.02   0.01   4.79
```

Rezultat poizvedbe je en sam dolg niz znakov. Zapis o vsakem genu je zaključen s treni poševnicami, ki mu sledi oznaka za prehod v novo vrstico. Delilni niz je zato `'///\n'`. S tem se tudi izognemo redundantni vrstici, ki jo dobimo pri delitvi v razdelku `??`.

```
> seznam <- strsplit(KEGGstr, "///\n")[[1]]
```

Iz vektorja z opisi izluščimo kode NCBI-GeneID

```
> GeneID <- sub("^.*NCBI-GeneID\\W*(\\w*).*$", "\\1", seznam,
+   extended = TRUE)
> Entry <- sub("^.*ENTRY\\W*(\\w*).*$", "\\1", seznam,
+   extended = TRUE)
> tail(cbind(Entry, GeneID))
   Entry  GeneID
[86,] "b3862" "948350"
[87,] "b1275" "945771"
[88,] "b2791" "947242"
[89,] "b1942" "946454"
[90,] "b1343" "947153"
[91,] "b4558" "1450304"
```

Izpis sekvence

```
> seq <- sub("^.*NTSEQ\\W*(\\w*).*$", "\\1", seznam[1],
+   extended = TRUE)
> cat(seq)
1299
```

```
atggctgaaagtactgtaacggcagacagcaaaactgacaagtagtgatactcgtcgccgc
at ttgggcgattgtggggcctcttcaggtaatctggcagtggttcgatttctatgct
tactcgttctgttcactctactttgccacatcttctccctccggaacacgacgact
caactactacaaacagcaggtgtttttgctgcgggattcctgatgcccgaataggcgg
tggctat ttggccgcatagccgataaacatggcgcacaaaatcgatgctgttatcggg
tgtatgatgtgtttcggatcgctggttatcgccctgcctcccagggttatgaaactatagg
acgtgggctccggcattattgcttctcgctcgtttatttcagggttatctgttggcgga
gaat atggcaccagcgcacacat atgagtgaagttgccgttgaagggcgcaaaggtttt
tacgcatcatttcagtatgtgacgcttgatcggcggacaactgctagccctactggttgc
gtggttttacaacacaccatggaagacgctgcactcagagagtggggatggcgtattcct
```

```

ttcgcgtaggagctgtgtagctggtgtggcggttggttacgtcgtcagttagatgaa
acttcgcaacaagaacgcgcgctttaaagaagctggatctctgaaaggattatggcgc
aatcgccgtgcattcatcatgggtctcggtttaccgctgcgggctccctttgttctat
accttactacttatatgcagaagtatctggtaaatactgcgggaatgcatgccaacgtg
gcgagtggcattatgactgccgcattggttgtattcatgcttattcaaccactcattggc
gcgctgtcggataagattggtcgccgtacctcaatggtatgtttcgggttcgctggcagcc
atTTTTaccgcttctattctctcagcattgcaaacgtttctcgccttatgccgctttt
ggtctggtgatgtgtgccctgctgatagtgagttttatacatcaatcagtggaatactg
aaggctgagatggtcccggcacaggttcgcgcattaggcggttggtctgtcatatgcggtc
gctaatgctatatttggtggttcggcggagtagtagcgttgtcgctgaaatcaatagga
atggaacagccttcttctggatgtgacctgatggccgtggtggcgttctggttctt
ttgatgctacatcgcaaaggaaggggatgcgtctttag

```

## 2 Luščenje posameznih označenih kod

S funkcijo `getTaggedCode`

```

> getTaggedCode <- function(x, tag = sub("^\\W*(\\w*).*$",
+ "\\1", x[1]), nocode = NA, ...) {
+   reg <- paste("^.*", tag, "\\W*(\\w*).*$", sep = "")
+   code <- sub(reg, "\\1", x, extended = TRUE, ...)
+   code[-grep(tag, x)] <- nocode
+   return(code)
+ }
> x <- c("ENTRY: 1111 XXX: x001", "ENTRY 22 XXX x02 YYY: Y01",
+ "ENTRY 3 YYY: Y03")
> cat(x, sep = "\n")
ENTRY: 1111 XXX: x001
ENTRY 22 XXX x02 YYY: Y01
ENTRY 3 YYY: Y03
> getTaggedCode(x, "XXX")
[1] "x001" "x02" NA
> getTaggedCode(x)
[1] "1111" "22" "3"

```

lahko izluščimo posamezne kode:

```

> tail(getTaggedCode(seznam, "GeneID"))
[1] "948350" "945771" "947242" "946454" "947153" "1450304"
> tail(getTaggedCode(gName, "Locus"))
[1] "b3862" "b1275" "b2791" "b1942" "b1343" "b4558"

```

Več o uporabi regularnih vzorcev za iskanje imenovanih kod je v datoteki [How-ToGetTaggedCodes.pdf](#).

Ker v KEGG niso navedeni vsi lokusi, moramo biti pri prireditvah previdni. V našem primeru nismo dobili informacije o vseh lokusih:

```

> length(Locus)
[1] 92
> length(Entry)
[1] 91
> notListed <- which(!Locus %in% Entry)
> notListed
[1] 25
> Locus[notListed]
[1] "b4274"

```

## 2.1 KEGG FTP

Na *FTP* strani <ftp://ftp.genome.jp/pub/kegg/genes/organisms/eco> pa najdemo šifrant za pretvorbo LocusID v NCBI-GeneID za *E. coli* `eco_ncbi-geneid.list`. To je seveda za prevedbo vseh genov precej ugodnejša varianta. Iz datoteke `eco_ncbi-geneid.list` bomo prepisali oznake v `featureData` za naše podatke.

## 2.2 Direktna povezava na KEGG

Kratek seznam genov, ki ima internetne povezave na bazo KEGG Genes:

```
> ID <- Locus[1:5]
> KEGG <- "http://www.genome.jp/dbget-bin/www_bget?eco"
> links <- paste(KEGG, ID, sep = "+")
> refs <- paste("\href{", links, "}{" , ID, "}", sep = "",
+             collapse = "\\\ \n")
> cat(refs, "\n")
```

[b2587](#)

[b1947](#)

[b0454](#)

[b1603](#)

[b0368](#)

Klik na ime pokaže podatke na spletni strani KEGG,  
<ctrl>-klik pripravi stran s podatki v obliki PDF datoteke  
<shift>-klik pa pripne PDF izpis spletne strani na konec tega dokumenta.



## SessionInfo

Windows XP (build 2600) Service Pack 3

- R version 2.10.0 (2009-10-26), i386-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: affy 1.24.0, annotate 1.24.0, AnnotationDbi 1.7.20, Biobase 2.6.0, bitops 1.0-4.1, DBI 0.2-4, Hmisc 3.7-0, KEGGSOAP 1.19.1, lumi 1.11.7, MASS 7.3-3, mgcv 1.5-6, preprocessCore 1.7.9, RCurl 1.2-1, RSQLite 0.7-3, survival 2.35-7
- Loaded via a namespace (and not attached): affyio 1.13.5, cluster 1.12.1, grid 2.10.0, lattice 0.17-26, nlme 3.1-96, SSOAP 0.5-4, tools 2.10.0, XML 2.6-0, XMLSchema 0.1-4, xtable 1.5-5