

The query language

Lobry, J.R.

June 11, 2014

Contents

1	Where to find information	2
2	Case sensitivity and ambiguities resolution	2
3	Selection criteria	2
3.1	Introduction	2
3.2	SP=taxon	3
3.3	TID=id	3
3.4	K=keyword	4
3.5	T=type	4
3.6	J=journal_name	4
3.7	R=refcode	5
3.8	AU=name	5
3.9	AC=accession_no	5
3.10	N=seq_name	6
3.11	NS=taxon_name	7
3.12	NK=keyword_name	7
3.13	Y=year or Y>year or Y<year	8
3.14	O=organelle	8
3.15	M=molecule	9
3.16	ST=status	9
3.17	F=file_name	10
3.18	FA=file_name	10
3.19	FK=file_name	11
3.20	FS=file_name	12
3.21	list_name	12
4	Operators	13
4.1	AND	13
4.2	OR	13
4.3	NOT	13
4.4	PAR	13
4.5	SUB	14

4.6	PS	14
4.7	PK	14
4.8	UN	14
4.9	SD	15
4.10	KD	15

References	17
------------	----

1 Where to find information

The last version of the documentation for the query language is available online at <http://pbil.univ-lyon1.fr/databases/acnuc/cfonctions.html#QUERYLANGUAGE>. This documentation has been imported within the documentation of the `query()` function, but the last available update is the online version. The query language is a specificity of the ACNUC system [5, 3, 4, 2].

2 Case sensitivity and ambiguities resolution

The query language is case insensitive, for instance:

```
choosebank("emblTP")
query("lowercase", "sp=escherichia coli", virtual = TRUE)
query("uppercase", "SP=Escherichia coli", virtual = TRUE)
lowercase$nelem == uppercase$nelem
```

```
[1] TRUE
```

```
closebank()
```

Three operators (AND, OR, NOT) can be ambiguous because they can also occur within valid criterion values. Such ambiguities can be solved by encapsulating elementary selection criteria between escaped double quotes. For example:

```
choosebank("emblTP")
query("ambig", "\"sp=Beak and feather disease virus\" AND \"au=ritchie\"", virtual = T)
ambig$nelem
```

```
[1] 18
```

```
closebank()
```

3 Selection criteria

3.1 Introduction

Selection criteria are in the form `c=something` (without space before the `=` sign) or `list_name` where `list_name` is a previously constructed list.

3.2 SP=taxon

This is used to select sequences attached to a given taxon or any other below in the tree. The at sign @ substitutes as a wildcard character for any zero or more characters. Here are some examples:

```
choosebank("emblTP")
query("bb", "sp=Borrelia burgdorferi", virtual=T)
bb$nelem
[1] 1682
query("borrelia", "sp=Borrelia", virtual=T)
borrelia$nelem
[1] 3173
closebank()
```

Here is an example of use of the wildcard @ to look for sapiens species:

```
choosebank("emblTP")
query("sapiens", "sp=@sapiens@", virtual=T)
sapiens$nelem
[1] 2216556
query("sapiensspecies", "PS sapiens")
getName(sapiensspecies)
[1] "HOMO SAPIENS"
[2] "HOMO SAPIENS NEANDERTHALENSIS"
[3] "HOMO SAPIENS X HUMAN PAPILLOMAVIRUS TYPE"
[4] "HOMO SAPIENS X SIMIAN VIRUS 40"
[5] "HOMO SAPIENS X HUMAN ENDOGENOUS RETROVIR"
[6] "HOMO SAPIENS X HUMAN T-CELL LYMPHOTROPIC"
[7] "HEPATITIS B VIRUS X HOMO SAPIENS"
[8] "HOMO SAPIENS X HEPATITIS B VIRUS"
[9] "HOMO SAPIENS X HUMAN IMMUNODEFICIENCY VI"
[10] "SYNTHETIC CONSTRUCT X HOMO SAPIENS"
[11] "HUMAN PAPILLOMAVIRUS X HOMO SAPIENS"
[12] "MUS SP. X HOMO SAPIENS"
[13] "HOMO SAPIENS X HUMAN PAPILLOMAVIRUS"
[14] "HOMO SAPIENS X HUMAN ADENOVIRUS TYPE 5"
[15] "HOMO SAPIENS X HERV-H/ENV62"
[16] "HOMO SAPIENS X HERV-H/ENV60"
[17] "HOMO SAPIENS X HERV-H/ENV59"
[18] "EXPRESSION VECTOR PTH-HIN X HOMO SAPIENS"
[19] "ADENO-ASSOCIATED VIRUS 2 X HOMO SAPIENS"
[20] "SIMIAN VIRUS 40 X HOMO SAPIENS"
[21] "HOMO SAPIENS X MUS MUSCULUS"
[22] "HOMO SAPIENS X INFLUENZA B VIRUS (B/LEE/"
[23] "MUS MUSCULUS X HOMO SAPIENS"
[24] "CRICETULUS GRISEUS X HOMO SAPIENS"
[25] "TRYPANOSOMA CRUZI X HOMO SAPIENS"
[26] "HOMO SAPIENS X TRYPANOSOMA CRUZI"
closebank()
```

3.3 TID=id

This is used to select sequences attached to a given numerical NCBI's taxonomy ID. For instance, the taxonomy ID for *Homo sapiens neanderthalensis* is 63221:

```
choosebank("genbank")
query("hsn", "TID=63221", virtual=T)
hsn$nelem
[1] 1355
```



Homo neanderthalensis.
Source: wikipedia

```

query("hsnsp", "PS hsn")
getName(hsnsp)
[1] "HOMO SAPIENS NEANDERTHALENSIS"
closebank()

```

3.4 K=keyword

This is used to select sequences attached to a given keyword or any other below in the tree. The at sign @ substitutes as a wildcard character for any zero or more characters. Example:

```

choosebank("emblTP")
query("ecoliribprot", "sp=escherichia coli AND k=rib@ prot@", virtual=T)
ecoliribprot$nelem
[1] 105
closebank()

```

3.5 T=type

This is used to select sequences of specified type. The list of available type for the currently opened database is given by function `getType()`:

```

choosebank("emblTP")
getType()

```

	sname	libel
2661	CDS	.PE protein coding region
2662	ID	Locus entry
2663	MISC_RNA	.RN other structural RNA coding region
2664	RRNA	.RR Ribosomal RNA coding gene
2665	SCRNA	.SC small cytoplasmic RNA
2666	SNRNA	.SN small nuclear RNA
2667	TRNA	.TR Transfer RNA coding gene

```

closebank()

```

For instance, to select all coding sequences from *Homo sapiens* we can use:

```

choosebank("emblTP")
query("hscds", "sp=Homo sapiens AND t=cds", virtual=T)
hscds$nelem
[1] 150513
closebank()

```

3.6 J=journal_name

This is used to select sequences published in journal specified using defined journal code. For instance to select all sequences published in *Science*:

```

choosebank("emblTP")
query("allseqsfromscience", "J=Science", virtual=TRUE)
allseqsfromscience$nelem
[1] 930397
closebank()

```

The list of available journal code can be obtained from the `readsmj()` function this way:

```
choosebank("emblTP")
nl <- readfirstrec(type = "SMJ")
smj <- readsmj(nl = nl, all.add = TRUE)
head(smj[!is.na(smj$nature) & smj$nature == "journal", c("sname", "libel")])
      sname                                libel
21      ABP                                Acta Biochim. Pol.
22 ABSTR-SOCNEUROSCI                      Abstr. - Soc. Neurosci.
23 ABSTRGENMEETAMSOCS                     Abstr. Gen. Meet. Am. Soc. Microbiol.
24 ABSTRMIDWINTERRESM Abstr. Midwinter Res. Meet. Assoc. Res. Otolaryngol.
25 ACTAAGRICSCANDAANI                     Acta Agric. Scand. A Anim. Sci.
26 ACTABIOCHIMBIOPHYS                     Acta Biochim. Biophys. Sin.

closebank()
```

3.7 R=refcode

This is used to select sequences from a given bibliographical reference specified as `jcode/volume/page`. For instance, to select sequences associated with the first publication [1] of the complete genome of *Rickettsia prowazekii*, we can use:

```
choosebank("emblTP")
query("rpro", "R=Nature/396/133")
getName(rpro)
[1] "RPDNAOMP" "RPXX01" "RPXX02" "RPXX03" "RPXX04"

closebank()
```

3.8 AU=name

This is used to select sequences having a specified author (only last name, no initial).

```
choosebank("emblTP")
query("Graur", "AU=Graur")
Graur$nelem
[1] 48

closebank()
```

3.9 AC=accession_no

This is used to select sequences attached to specified accession number. For instance if we are looking for sequences attached to the accession number AY382159:

```
choosebank("emblTP")
query("ACexample", "AC=AY382159")
getName(ACexample$req[[1]])
[1] "AY382159"

annotations <- getAnnot(ACexample$req[[1]])
cat(annotations, sep = "\n")
```

```

ID   AY382159   standard; genomic DNA; PRO; 783 BP.
XX
AC   AY382159;
XX
SV   AY382159.1
XX
DT   08-OCT-2003 (Rel. 77, Created)
DT   08-OCT-2003 (Rel. 77, Last updated, Version 1)
XX
DE   Borrelia burgdorferi strain FP1 OspA gene, partial cds.
XX
KW   .
XX
OS   Borrelia burgdorferi (Lyme disease spirochete)
OC   Bacteria; Spirochaetes; Spirochaetales; Spirochaetaceae; Borrelia;
OC   Borrelia burgdorferi group.
XX
RN   [1]
RP   1-783
RA   Hao Q., Wan K.;
RT   ;
RL   Submitted (03-SEP-2003) to the EMBL/GenBank/DBJ databases.
RL   Department of Lyme Spirochetosis, CDC, Beijing 102206, China
XX
FH   Key          Location/Qualifiers
FH
FT   source          1..783
FT                   /db_xref="taxon:139"
FT                   /mol_type="genomic DNA"
FT                   /organism="Borrelia burgdorferi"
FT                   /strain="FP1"
FT   CDS              <1..>783
FT                   /codon_start=1
FT                   /transl_table=11
FT                   /product="OspA"
FT                   /protein_id="AAQ89576.1"
FT                   /translation="ALIACKQNVSSLDEKNSASVDLPGEMKVLVSKEKDKDGKYSLKAT
FT                   VDKLELKGTSDKNNGSGTLEGEKTDKSKAKLTISDDLKTTFEVFKEDGKTLVSRKVSS
FT                   KDKTSTDEMFNEKGELSAKMTRENGTKLEYTEMKSDGTGKTKEVLKNFTLEGRVANDK
FT                   VTLEVKEGTVTLKSKEIAKSGEVTVALNDTNTTQATKKTGAWDSKSTLTISVNSKKTQ
FT                   LVFTKQDTITVQKYSAGTNLEGTAVEIKTLDELKNALK"
XX
SQ   Sequence 783 BP; 342 A; 124 C; 145 G; 172 T; 0 other;

closebank()

```

3.10 N=seq_name

This is used to select sequences of a given name¹. Sequences names are not necessarily stable, so that it's almost always better to work with accession numbers. Anyway, the distinction between sequence names and accession numbers is on a vanishing way because they tend more and more to be the same thing (as in the example just below). The use of the at sign @ to substitute as a wildcard character for any zero or more characters is possible here.

```

choosebank("emblTP")
query("Nexample", "N=AY382159")
getName(Nexample$req[[1]])

[1] "AY382159"

annotations <- getAnnot(Nexample$req[[1]])
cat(annotations, sep = "\n")

```

¹*i.e.* what is documented in the ID or the LOCUS field

```

ID  AY382159   standard; genomic DNA; PRO; 783 BP.
XX
AC  AY382159;
XX
SV  AY382159.1
XX
DT  08-OCT-2003 (Rel. 77, Created)
DT  08-OCT-2003 (Rel. 77, Last updated, Version 1)
XX
DE  Borrelia burgdorferi strain FP1 OspA gene, partial cds.
XX
KW  .
XX
OS  Borrelia burgdorferi (Lyme disease spirochete)
OC  Bacteria; Spirochaetes; Spirochaetales; Spirochaetaceae; Borrelia;
OC  Borrelia burgdorferi group.
XX
RN  [1]
RP  1-783
RA  Hao Q., Wan K.;
RT  ;
RL  Submitted (03-SEP-2003) to the EMBL/GenBank/DDBJ databases.
RL  Department of Lyme Spirochetosis, CDC, Beijing 102206, China
XX
FH  Key          Location/Qualifiers
FH
FT  source        1..783
FT                /db_xref="taxon:139"
FT                /mol_type="genomic DNA"
FT                /organism="Borrelia burgdorferi"
FT                /strain="FP1"
FT  CDS            <1..>783
FT                /codon_start=1
FT                /transl_table=11
FT                /product="OspA"
FT                /protein_id="AAQ89576.1"
FT                /translation="ALIAACKQNVSSLDEKNSASVDLPGEMKVLVSKEKDKDGKYSKAT
FT                VDKLELKGTSKNNNGSGTLEGEKTDKSKAKLTISDDLKTTFEVFKEDGKTLVSRKVSS
FT                KDKTSTDEMFNEKGELSAKTMTRENGTKLEYTEMKSDGTGKTKVLEKNFTLEGRVANDK
FT                VTLEVKEGTVTSLKEIAKSCEVTVALNDTNTTQATKKTGAWDSKSTLTISVNSKKTQ
FT                LVFTKQDTITVQKYSAGTNLEGTAVEIKTLDELKNALK"
XX
SQ  Sequence 783 BP; 342 A; 124 C; 145 G; 172 T; 0 other;
closebank()

```

3.11 NS=taxon_name

This is used to get the number of taxon of given name, with the use of the at sign @ to substitute as a wildcard character for any zero or more characters possible here. For instance, we want to know how many taxon have *sapiens* inside :

```

choosebank("emblTP")
query("NSexample", "NS=@sapiens@")
NSexample
26 SP for NS=@sapiens@
closebank()

```

3.12 NK=keyword_name

This is used to get the number of keyword of given name, with the use of the at sign @ to substitute as a wildcard character for any zero or more characters

possible here. For instance, we want to know how many keywords have *sex* inside :

```
choosebank("emblTP")
query("NKexample", "NK=@sex@")
NKexample
277 KW for NK=@sex@
closebank()
```

3.13 Y=year or Y>year or Y<year

This is used to select sequences published in a given year (**Y=year**), or in a given year and after this year (**Y>year**), or in a given year and before this year (**Y<year**).

```
choosebank("emblTP")
query("Yexample", "Y=1999", virtual=TRUE)
Yexample$nelem
[1] 955274
closebank()
```

3.14 O=organelle

This is used to select sequences from specified organelle named following defined code (*e.g.*, chloroplast). The list of available organelle codes can be obtained from the `readsmj()` function this way:

```
choosebank("genbank")
nl <- readfirstrec(type = "SMJ")
smj <- readsmj(nl = nl, all.add = TRUE)
smj[!is.na(smj$nature) & smj$nature == "organelle", c("sname", "libel")]
      sname          libel
5278  CHLOROPLAST      Chloroplast genome
5279  CHROMATOPHORE      <NA>
5280  HYDROGENOSOME      <NA>
5281  MITOCHONDRION      Mitochondrial genome
5282  NUCLEOMORPH        Nucleomorph genome
5283  PLASTID non-green plastid genome
closebank()
```

To select for instance all sequences from chloroplast genome we can use:

```
choosebank("emblTP")
query("Oexample", "O=chloroplast", virtual=TRUE)
Oexample$nelem
[1] 65011
closebank()
```


3.15 M=molecule

This is used to select sequences according to the chemical nature of the sequenced molecule². The list of available organelle code can be obtained from the `readsmj()` function this way:

```
choosebank("genbank")
nl <- readfirstrec(type = "SMJ")
smj <- readsmj(nl = nl, all.add = TRUE)
smj[!is.na(smj$nature) & smj$nature == "molecule", c("sname", "libel")]

  sname                                libel
4  CRNA Sequenced molecule is complementary RNA
5  DNA      Sequenced molecule is DNA
6  MRNA      sequenced molecule is mRNA
7  RNA       Sequenced molecule is RNA
8  RRNA      sequenced molecule is rRNA
9  TRNA      sequenced molecule is tRNA

closebank()
```

To select for instance all sequences sequenced from DNA we can use:

```
choosebank("emblTP")
query("Mexample", "M=DNA", virtual=TRUE)
Mexample$nelem
[1] 7421752
closebank()
```

3.16 ST=status

This is used to select sequences from specified data class (EMBL) or review level (UniProt). The list of status codes can be obtained from the `readsmj()` function this way:

```
choosebank("embl")
nl <- readfirstrec(type = "SMJ")
smj <- readsmj(nl = nl, all.add = TRUE)
smj[!is.na(smj$nature) & smj$nature == "status", c("sname", "libel")]

  sname                                libel
1  ANN      Annotated CON data class
2  EST      Expressed Sequence Tags data class
3  GSS      Genome Survey Sequence data class
4  HTC      High Throughput cDNA data class
5  HTG High Throughput Genome sequencing data class
6  PAT      Patent data class
7  STD      standard data class
8  STS      Sequence Tagged Site data class
9  TPA      Third Party Annotation data class
10 TSA      Transcriptome Shotgun Assembly data class

closebank()
choosebank("swissprot")
nl <- readfirstrec(type = "SMJ")
smj <- readsmj(nl = nl, all.add = TRUE)
smj[!is.na(smj$nature) & smj$nature == "status", c("sname", "libel")]

  sname                                libel
1  REVIEWED Entry was reviewed and annotated by UniProtKB curators
2  UNREVIEWED      Computer-annotated entry

closebank()
```

²as named in ID or LOCUS annotation records

To select for instance all fully annotated sequences from Uniprot we can use:

```
choosebank("swissprot")
query("STexample","ST=REVIEWED", virtual=TRUE)
STexample$nelem
[1] 545388
closebank()
```

3.17 F=file_name

This is used to select sequences whose names are in a given file, one name per line. This is not directly implemented in seqinR, you have to use the function `crelistfromclientdata()` or its short form `clfcd()` for this purpose. Here is an example with a file of sequence names distributed with the seqinR package:

```
choosebank("emblTP")
fileSQ <- system.file("sequences/bb.mne", package = "seqinr")
cat(readLines(fileSQ),sep="\n")
A04009.OSPA
A04009.OSPB
A22442
A24006
A24008
A24010
A24012
A24014
A24016
A33362
A67759.PE1
AB011063
AB011064
AB011065
AB011066
AB011067
AB035616
AB035617
AB035618
AB041949.VLSE

clfcd("listSQ", file = fileSQ, type = "SQ")
getName(listSQ)
[1] "A04009.OSPA"    "A04009.OSPB"    "A22442"         "A24006"
[5] "A24008"         "A24010"         "A24012"         "A24014"
[9] "A24016"         "A33362"         "A67759.PE1"     "AB011063"
[13] "AB011064"       "AB011065"       "AB011066"       "AB011067"
[17] "AB035616"       "AB035617"       "AB035618"       "AB041949.VLSE"

closebank()
```

3.18 FA=file_name

This is used to select sequences whose accession numbers are in a given file, one name per line. This is not directly implemented in seqinR, you have to use the function `crelistfromclientdata()` or its short form `clfcd()` for this purpose. Here is an example with a file of sequence accession numbers distributed with the seqinR package:

```
choosebank("emblTP")
fileAC <- system.file("sequences/bb.acc", package = "seqinr")
cat(readLines(fileAC),sep="\n")
```

```

AY382159
AY382160
AY491412
AY498719
AY498720
AY498721
AY498722
AY498723
AY498724
AY498725
AY498726
AY498727
AY498728
AY498729
AY499181
AY500379
AY500380
AY500381
AY500382
AY500383

  clfcdb("listAC", file = fileAC, type = "AC")
  getName(listAC)

[1] "AY382159" "AY382160" "AY491412" "AY498719" "AY498720" "AY498721"
[7] "AY498722" "AY498723" "AY498724" "AY498725" "AY498726" "AY498727"
[13] "AY498728" "AY498729" "AY499181" "AY500379" "AY500380" "AY500381"
[19] "AY500382" "AY500383"

  closebank()

```

3.19 FK=file_name

This is used to produce the list of keywords named in given file, one keyword per line. This is not directly implemented in seqinR, you have to use the function `crelistfromclientdata()` or its short form `clfcdb()` for this purpose. Here is an example with a file of keywords distributed with the seqinR package:

```

choosebank("emblTP")
fileKW <- system.file("sequences/bb.kwd", package = "seqinr")
cat(readLines(fileKW), sep="\n")

PLASMID
CIRCULAR
PARTIAL
5'-PARTIAL
3'-PARTIAL
MOTA GENE
MOTB GENE
DIVISION PRO
GYRB GENE
JOINING REGION
FTSA GENE
RPOB GENE
RPOC GENE
FLA GENE
DNAJ GENE
TUF GENE
PGK GENE
RUVA GENE
RUVB GENE
PROMOTER REGION

  clfcdb("listKW", file = fileKW, type = "KW")
  getName(listKW)

[1] "PLASMID"          "CIRCULAR"          "PARTIAL"           "5'-PARTIAL"
[5] "3'-PARTIAL"       "MOTA GENE"         "MOTB GENE"         "DIVISION PRO"
[9] "GYRB GENE"        "JOINING REGION"    "FTSA GENE"         "RPOB GENE"
[13] "RPOC GENE"        "FLA GENE"          "DNAJ GENE"         "TUF GENE"
[17] "PGK GENE"         "RUVA GENE"         "RUVB GENE"         "PROMOTER REGION"

  closebank()

```

3.20 FS=file_name

This is used to produce the list of species named in given file, one species per line. This is not directly implemented in seqinR, you have to use the function `crelistfromclientdata()` or its short form `clfcd()` for this purpose. Here is an example with a file of species names distributed with the seqinR package:

```
choosebank("emblTP")
fileSP <- system.file("sequences/bb.sp", package = "seqinr")
cat(readLines(fileSP), sep="\n")

BORRELIA ANSERINA
BORRELIA CORIACEAE
BORRELIA PARKERI
BORRELIA TURICATAE
BORRELIA HERMSII
BORRELIA CROCIDURAE
BORRELIA LONESTARI
BORRELIA HISPANICA
BORRELIA BARBOURI
BORRELIA THEILERI
BORRELIA DUTTONII
BORRELIA MIYAMOTOI
BORRELIA PERSICA
BORRELIA RECURRENTIS
BORRELIA BURGDORFERI
BORRELIA AFZELII
BORRELIA GARINII
BORRELIA ANDERSONII
BORRELIA VALAISIANA
BORRELIA JAPONICA

clfcd("listSP", file = fileSP, type = "SP")
getName(listSP)

[1] "BORRELIA ANSERINA"      "BORRELIA CORIACEAE"      "BORRELIA PARKERI"
[4] "BORRELIA TURICATAE"     "BORRELIA HERMSII"        "BORRELIA CROCIDURAE"
[7] "BORRELIA LONESTARI"     "BORRELIA HISPANICA"      "BORRELIA BARBOURI"
[10] "BORRELIA THEILERI"      "BORRELIA DUTTONII"       "BORRELIA MIYAMOTOI"
[13] "BORRELIA PERSICA"       "BORRELIA RECURRENTIS"    "BORRELIA BURGDORFERI"
[16] "BORRELIA AFZELII"       "BORRELIA GARINII"        "BORRELIA ANDERSONII"
[19] "BORRELIA VALAISIANA"    "BORRELIA JAPONICA"

closebank()
```

3.21 list_name

A list name can be re-used, for instance:

```
choosebank("emblTP")
query("MyFirstListName", "Y=2000", virtual = TRUE)
MyFirstListName$nelem

[1] 885225

query("MySecondListName", "SP=Borrelia burgdorferi", virtual = TRUE)
MySecondListName$nelem

[1] 1682

query("MyThirdListName", "MyFirstListName AND MySecondListName", virtual = TRUE)
MyThirdListName$nelem

[1] 131

closebank()
```

4 Operators

4.1 AND

This is the binary operator for the logical and: a sequence belongs to the resulting list if, and only if, it is present in both operands. To select for instance sequences from *Borrelia burgdorferi* that are also coding sequences we can use:

```
choosebank("emblTP")
query("ANDexample", "SP=Borrelia burgdorferi AND T=CDS", virtual=TRUE)
ANDexample$nelem
[1] 3218
closebank()
```

4.2 OR

This is the binary operator for the logical or: a sequence belongs to the resulting list if it is present in at least one of the two operands. To select for instance sequences from *Borrelia burgdorferi* or from *Escherichia coli* we can use:

```
choosebank("emblTP")
query("ORexample", "SP=Borrelia burgdorferi OR SP=Escherichia coli", virtual=TRUE)
ORexample$nelem
[1] 28584
closebank()
```

4.3 NOT

This is the unary operator for the logical negation. To select for instance sequences from *Borrelia burgdorferi* that are not partial we can use:

```
choosebank("emblTP")
query("NOTexample", "SP=Borrelia burgdorferi AND NOT K=PARTIAL", virtual=TRUE)
NOTexample$nelem
[1] 3266
closebank()
```

4.4 PAR

This is a unary operator to compute the list of parent sequences of a list of sequences. The reciprocal operator is SUB. To check the reciprocity we can use for instance:

```
choosebank("emblTP")
query("A", "T=TRNA", virtual=TRUE)
query("B", "PAR A", virtual=TRUE)
query("C", "SUB B", virtual=TRUE)
query("D", "PAR C", virtual=TRUE)
query("emptySet", "B AND NOT D", virtual=TRUE)
emptySet$nelem
[1] 0
closebank()
```

4.5 SUB

This is a unary operator to add all subsequences of members of the single list operand.

```
choosebank("emblTP")
query("SUBexample", "AC=AE000783", virtual=T)
SUBexample$nelem
[1] 70
query("SUBexample2", "SUB SUBexample", virtual=T)
SUBexample2$nelem
[1] 943
closebank()
```

4.6 PS

This unary operator is used to get the list of species attached to member sequences of the operand list.

```
choosebank("emblTP")
query("PSexample", "K=hyperthermo@", virtual=T)
query("PSexample2", "PS PSexample")
getName(PSexample2)
[1] "BACILLUS LICHENIFORMIS" "DESULFUROCOCCUS"
[3] "PYROCOCCUS FURIOSUS"
closebank()
```

4.7 PK

This unary operator is used to get the list of keywords attached to member sequences of the operand list.

```
choosebank("emblTP")
query("PKexample", "AC=AE000783", virtual=T)
query("PKexample2", "PK PKexample")
getName(PKexample2)
[1] "DIVISION PRO" "CDS" "RRNA" "TRNA"
[5] "SOURCE" "RELEASE 75"
closebank()
```

4.8 UN

This unary operator is used to get the list of sequences attached to a list of species or keywords.

```
choosebank("emblTP")
fileSP <- system.file("sequences/bb.sp", package = "seqinr")
cat(readLines(fileSP), sep="\n")
```

```

BORRELIA ANSERINA
BORRELIA CORIACEAE
BORRELIA PARKERI
BORRELIA TURICATAE
BORRELIA HERMSII
BORRELIA CROCIDURAE
BORRELIA LONESTARI
BORRELIA HISPANICA
BORRELIA BARBOURI
BORRELIA THEILERI
BORRELIA DUTTONII
BORRELIA MIYAMOTOI
BORRELIA PERSICA
BORRELIA RECURRENTIS
BORRELIA BURGDORFERI
BORRELIA AFZELII
BORRELIA GARINII
BORRELIA ANDERSONII
BORRELIA VALAISTIANA
BORRELIA JAPONICA

clfcd("listSP", file = fileSP, type = "SP")
query("UNexample", "UN listSP", virtual = TRUE)
UNexample$nelem

[1] 2877

closebank()

```

4.9 SD

This unary operator computes the list of species placed in the tree below the members of the species list operand.

```

choosebank("emblTP")
query("hominidae", "SP=Hominidae", virtual=T)
query("hsp", "PS hominidae", virtual=T)
hsp$nelem

[1] 19

query("SDexample", "SD hsp")
getName(SDexample)

[1] "HOMINIDAE"                "PONGO"
[3] "PONGO PYGMAEUS"           "PONGO PYGMAEUS ABELII"
[5] "PONGO PYGMAEUS PYGMAEUS"  "PONGO SP."
[7] "HOMO/PAN/GORILLA GROUP"   "GORILLA"
[9] "GORILLA GORILLA"          "GORILLA GORILLA BERINGEI"
[11] "GORILLA GORILLA GRAUERI"  "GORILLA GORILLA GORILLA"
[13] "GORILLA GORILLA VELLENSIS" "PAN"
[15] "PAN TROGLODYTES"          "PAN TROGLODYTES SCHWEINFURTHII"
[17] "PAN TROGLODYTES TROGLODYTES" "PAN TROGLODYTES VERUS"
[19] "PAN TROGLODYTES VELLEROSUS" "PAN PANISCUS"
[21] "HOMO"                     "HOMO SAPIENS"
[23] "HOMO SAPIENS NEANDERTHALENSIS"

closebank()

```

4.10 KD

This unary operator computes the list of keywords placed in the tree below the members of the keywords list operand.


```

choosebank("emblTP")
query("cat", "SP=Felis catus", virtual = TRUE)
query("catkw", "PK cat", virtual = TRUE)
catkw$nelem

```


```
[1] 540
      query("KDexample", "KD catkw", virtual = TRUE)
      KDexample$nelem
[1] 572
      closebank()
```

Session Informations

This part was compiled under the following  environment:

- R version 3.1.0 (2014-04-10), x86_64-apple-darwin13.1.0
- Locale: fr_FR.UTF-8/fr_FR.UTF-8/fr_FR.UTF-8/C/fr_FR.UTF-8/fr_FR.UTF-8
- Base packages: base, datasets, graphics, grDevices, grid, methods, stats, utils
- Other packages: ade4 1.6-2, ape 3.1-2, grImport 0.9-0, MASS 7.3-31, seqinr 3.0-11, tseries 0.10-32, XML 3.98-1.1, xtable 1.7-3
- Loaded via a namespace (and not attached): lattice 0.20-29, nlme 3.1-117, quadprog 1.5-5, tools 3.1.0, zoo 1.7-11

There were two compilation steps:

-  compilation time was: Wed Jun 11 10:58:16 2014
- \LaTeX compilation time was: June 11, 2014

References

- [1] S.G. Andersson, A. Zomorodipour, J.O. Andersson, T. Sicheritz-Ponten, U.C. Alsmark, R.M. Podowski, A.K. Naslund, A.S. Eriksson, H.H. Winkler, and C.G. Kurland. The genome sequence of *Rickettsia prowazekii* and the origin of mitochondria. *Nature*, 396:133–140, 1998.
- [2] M. Gouy and S. Delmotte. Remote access to ACNUC nucleotide and protein sequence databases at PBIL. *Biochimie*, 90:555–562, 2008.
- [3] M. Gouy, C. Gautier, M. Attimonelli, C. Lanave, and G. di Paola. ACNUC—a portable retrieval system for nucleic acid sequence databases: logical and physical designs and usage. *Computer Applications in the Biosciences*, 1:167–172, 1985.
- [4] M. Gouy, C. Gautier, and F. Milleret. System analysis and nucleic acid sequence banks. *Biochimie*, 67:433–436, 1985.
- [5] 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.