Quick Guide for BioIDMapper Package

Xiaoyong Sun[†‡∗]

March 5, 2008

[†]Binformatics and Computational Biology Program, [‡]Department of Statistics
Iowa State University, Ames, Iowa 50010, USA

Contents

1 Introduction

Many new databases aiming at genes and proteins are developed as more and more species are sequenced. It becomes tedious job about how to navigate among different data resources, map various IDs, and analyze separate biological knowledge. Current popular databases include Entrez Gene, UniProt, Gene Ontology, EMBL, OMIM, PubMed, KEGG, etc. Based on NCBI, UniProt, KEGG and other web services, BioIDMapper can facilitate mapping between different databases, integrate various ID systems and provide a full view from

---

[∗]sunx1@iastate.edu

gene level, mRNA level and functional level regarding one specific ID. This package is based on NCBI and UniProt websites, utilizing two packages: XML and RCurl.

2 BioID mapping table

In this package, the following 59 BioIDs can be translated to each other.

Note:
(1)"GI number" shows in both NCBI and UniPort databases, and it serves as bridge between two databases.
(2) "Biokey number" is the "currency" connecting functions.

Biokey number	BioIDs	Sources
1	GI number	NCBI
2	Pubmed id	NCBI
3	GEO id	NCBI
4	OMIM id	NCBI
5	SNP id	NCBI
6	UniGene cluster id	NCBI
7	UniSTS id	NCBI
8	Popset id	NCBI
9	MMDB id	NCBI
10	3D SDI id	NCBI
11	PSSM id	NCBI
12	TAXID	NCBI
13	Genome id	NCBI
14	PubChem Compound id	NCBI
15	PubChem Substance id	NCBI
16	PubChem BioAssay id	NCBI
17	NNNNNN	Boundary
18	GI number	UniProt
19	UniProtKB Accession	UniProt
20	UniProtKB id	UniProt
21	PIR Accession	UniProt
22	Enzyme Commission	UniProt
23	GO id	UniProt
24	Entrez Gene id	UniProt
25	EMBL id	UniProt
26	ENSEMBL id	UniProt
27	UniGene id	UniProt
28	TAIR id	UniProt
29	TIGR id	UniProt
30	KEGG id	UniProt

Biokey number	BioIDs	Sources
31	NCBI Taxon id	UniProt
32	OMIM id	UniProt
33	Ecogene id	UniProt
34	Flybase id	UniProt
35	GENEDB_SPOMBE id	UniProt
36	GERMONLINE id	UniProt
37	GRAMENE id	UniProt
38	HIV id	UniProt
39	IPI	UniProt
40	PDB id	UniProt
41	REBASE id	UniProt
42	Refseq Accession	UniProt
43	SGD id	UniProt
44	TRANSFAC id	UniProt
45	WORMPEP id	UniProt
46	UniRef100 id	UniProt
47	UniRef90 id	UniProt
48	UniRef50 id	UniProt
49	InterPro id	UniProt
50	Medline id	UniProt
51	PFAM id	UniProt
52	PIRSF id	UniProt
53	PRINTS id	UniProt
54	PRODOM id	UniProt
55	PROSITE id	UniProt
56	PMID	UniProt
57	SMART id	UniProt
58	TAXGRPID	UniProt
59	TIGRFAMs id	UniProt
60	TRANSFAC id	UniProt

3 Function description

This package includes one standard mapping table displayed by bio.type()function, and three features: mapping, linking and data analysis. Mapping feature is implemented by bio.convert() function; linking feature involves bio.link() function, and data analysis is done by 2 functions: bio.sum() and bio.select().

3.1 Retrieve mapping table

3.1.1 bio.type

`bio.type <-function(type2id)`

Show all Biokey numbers, and biological types that this package can handle.
Presently 59 biological types are included in the package.
It takes one parameters:
a. type2id
Biokey number or BioIDs (biological types) from BioID mapping table;

If no argument is used, return BioID mapping table for all biokey numbers;
If argument is Biokey number from bio.type(), return the corresponding biological type;
If argument is biological type, return the corresponding Biokey number.

```
>bio.type()

     Biokey number BioIDs                Sources
 [1,] "1"          "GI number"          "NCBI"
 [2,] "2"          "Pubmed id"          "NCBI"
 [3,] "3"          "GEO id"             "NCBI"
 [4,] "4"          "OMIM id"            "NCBI"
 [5,] "5"          "SNP id"             "NCBI"
 [6,] "6"          "UniGene cluster id" "NCBI"
 [7,] "7"          "UniSTS id"          "NCBI"
 [8,] "8"          "Popset id"          "NCBI"
 [9,] "9"          "MMDB id"            "NCBI"
[10,] "10"         "3D SDI id"          "NCBI"
.....

> library(BioIDMapper)
> bio.type(5)

[1] "SNP id"

> bio.type("SNP id")

[1] 5
```

3.2 Mapping feature

3.2.1 bio.convert

```
bio.convert <-function(id_list, from, to)
```

This is the main interface for mapping ids. It takes three parameters:
a. id_list: id list you want to map;
b. from: Biokey number of source type; bio.type() will show all the Biokey numbers for biological types.
c. to: Biokey number of destination type; bio.type() will show all the Biokey numbers for biological types.

```
> data(glist)
> myMap <- bio.convert(glist, 1, 5)
> myMap[1:10, ]

  GI number snp
1 "200533"  "27242152"
2 "200533"  "27242151"
3 "200533"  "27242150"
```

```
4  "200533"  "27242149"
5  "200533"  "27242148"
6  "200533"  "27242147"
7  "200533"  "27242146"
8  "200533"  "27242145"
9  "200533"  "27242144"
10 "200533"  "27242143"
```

3.3 Linking feature

3.3.1 bio.link

```
bio.link <-function(id, to)
```

This is the main interface for linking to external data sources. It will start web
browser, and link that id to external data source. It takes two parameters:
a. id: id you want to link;
b. to: The corresponding Biokey number of external biology types you want to
link; bio.type() will show all Biokey number for biological types.
Note:
"id" should match "to". For example, id: "27242148" is "SNP id"; to: 5 is Biokey
number for "SNP id".

```
>bio.link("27242148", 5)
```

3.4 Data analysis feature

This feature is to analyze result from mapping function: bio.convert().

3.4.1 bio.sum

```
bio.sum <-function(result_matrix, start_idList, option)
```

Summary the result after mapping. It takes three parameters:
a. result_matrix: result matrix from bio.convert() function
b. start_idList: the orginial id list you want to map
c. option: a logical value. If TRUE, all summary results are returned. If FALSE,
only basic summary is returned. default value is FALSE

```
> data(glist)
> myMap <- bio.convert(glist, 1, 5)
> bio.sum(myMap)

----------------------------------------------------------------

MAPPING SCHEMA:
38 GI number  are mapped to  272 snp
----------------------------------------------------------------

$`Summary for result`
            GI number snp
maping_Total       38 272
```

```
> mySum <- bio.sum(myMap, glist, FALSE)

------------------------------------------------------------------

MAPPING SCHEMA:
38 GI number  are mapped to  272 snp

MAPPING PERCENTAGE:
100.00% GI number  are mapped to  snp
------------------------------------------------------------------
```

3.4.2 bio.select

```
bio.select <-function(myid, result_matrix, colno)
```

Show mapping result for one id. It takes three parameters:
a. myid: id you are interested
b. result_matrix: result matrix from bio.convert() function
c. colno: the column number of result_matrix that contains id you are interested.

```
> data(glist)
> myMap <- bio.convert(glist, 1, 5)
> bio.select(myMap, 1, "41386735")

    GI number  snp
609 "41386735" "43102393"
610 "41386735" "43102392"
611 "41386735" "43102391"
612 "41386735" "43102390"
613 "41386735" "43102389"
614 "41386735" "43102388"
615 "41386735" "43102387"
616 "41386735" "43102386"
617 "41386735" "43102385"
618 "41386735" "41694775"
619 "41386735" "41694774"
620 "41386735" "41694773"
621 "41386735" "41694772"
622 "41386735" "41694771"
623 "41386735" "41694770"
624 "41386735" "41694769"
625 "41386735" "41694768"
626 "41386735" "41694767"
627 "41386735" "41694766"
628 "41386735" "41694765"
```

4   Demonstration

To illustrate how to use this package, three examples are used, including mapping within NCBI, UniProt respectively, and mapping between NCBI and UniProt, to show how to translate different biological ids .

4.1 Examples for mapping within NCBI

Let's assume that you have 500 Genbank gi numbers, and you are interested in related snp ids.

First, you can find Biokey number from bio.type() function,

```
> bio.type()
```

```
      Biokey number BioIDs              Sources
 [1,] "1"           "GI number"         "NCBI"
 [2,] "2"           "Pubmed id"         "NCBI"
 [3,] "3"           "GEO id"            "NCBI"
 [4,] "4"           "OMIM id"           "NCBI"
 [5,] "5"           "SNP id"            "NCBI"
 [6,] "6"           "UniGene cluster id"  "NCBI"
 [7,] "7"           "UniSTS id"         "NCBI"
 [8,] "8"           "Popset id"         "NCBI"
 [9,] "9"           "MMDB id"           "NCBI"
[10,] "10"          "3D SDI id"         "NCBI"
.....
```

Second, you can use the bio.convert() to map from one type of id to the other type of id:

```
> data(glist)
> myMap <- bio.convert(glist, 1, 5)
> myMap[80:91, ]
```

```
   GI number snp
80 "200529"  "27242149"
81 "200529"  "27242148"
82 "200529"  "27242147"
83 "200529"  "27242146"
84 "200529"  "27242145"
85 "200529"  "27242144"
86 "200529"  "27242143"
87 "200529"  "27242142"
88 "200529"  "27242141"
89 "200529"  "27242140"
90 "200529"  "27242139"
91 "200529"  "27242138"
```

In addition, you can analyze the mapping result with bio.sum() function

```
>bio.sum(myMap)
```

```
          [,1]      [,2]
          "protein" "snp"
mappingNo "38"      "272"
```

Also, you can select the id you are interested from result:

```
>bio.select(myMap, 1, "200529")
```

Finally, you can check the detailed information about snp with id: "27242138" using bio.link(), and it will give you more detailed information from web browser.

```
>bio.link("27242138", 5)
```

4.2   Examples for mapping within UniProt

Assume that we have 10 UniProt Accession numbers, and let's find the related PDB ids.
First, you need to find out related id number in BioIDMapper package using bio.type function:

```
> bio.type()
.....
[19,] "19"       "UniProtKB Accession"  "UniProt"
[20,] "20"       "UniProtKB id"         "UniProt"
[21,] "21"       "PIR Accession"        "UniProt"
[22,] "22"       "Enzyme Commission"    "UniProt"
[23,] "23"       "Go id"                "UniProt"
[24,] "24"       "Entrez Gene id"       "UniProt"
[25,] "25"       "EMBL id"              "UniProt"
[26,] "26"       "ENSEMBL id"           "UniProt"
[27,] "27"       "UniGene id"           "UniProt"
[28,] "28"       "TAIR id"              "UniProt"
[29,] "29"       "TIGR id"              "UniProt"
[30,] "30"       "KEGG id"              "UniProt"
[31,] "31"       "NCBI Taxon id"        "UniProt"
[32,] "32"       "OMIM id"              "UniProt"
[33,] "33"       "Ecogene id"           "UniProt"
[34,] "34"       "Flybase id"           "UniProt"
[35,] "35"       "GENEDB_SPOMBE id"     "UniProt"
[36,] "36"       "GERMONLINE id"        "UniProt"
[37,] "37"       "GRAMENE id"           "UniProt"
[38,] "38"       "HIV id"               "UniProt"
[39,] "39"       "IPI"                  "UniProt"
[40,] "40"       "PDB id"               "UniProt"
.....

> bio.type("PDB id")

[1] 40
```

Second, you can use the bio.convert() to map from one type of id to the other type of id:

```
> data(ulist)
> myMap <- bio.convert(ulist, 19, 40)
> myMap[1:10, ]

   ACC       PDB_ID
1  "P04925" "1AG2"
2  "P04925" "1XYX"
```

```
3  "P04925" "1Y15"
4  "P04925" "1Y16-PDBSUM_ID-"
5  "Q4FJQ7" "1E1G"
6  "Q4FJQ7" "1E1J"
7  "Q4FJQ7" "1E1P"
8  "Q4FJQ7" "1E1S"
9  "Q4FJQ7" "1E1U"
10 "Q4FJQ7" "1E1W"
```

You can also utilize bio.sum(), bio.select(), and bio.link() tools to check the related information.

4.3   Examples for mapping between NCBI and UniProt

You can do the mapping between NCBI and Uniprot exactly as before. Currently the bridge is "GI number" between the mapping of NCBI and Uniprot.

If you are interested in translating UniProt Accession Number to SNP id,

```
> data(ulist)
> myMap <- bio.convert(ulist, 19, 5)
> myMap[1:10, ]

      ACC    P_GI       snp
1   P04925 200533 27242152
2   P04925 200533 27242151
3   P04925 200533 27242150
4   P04925 200533 27242149
5   P04925 200533 27242148
6   P04925 200533 27242147
7   P04925 200533 27242146
8   P04925 200533 27242145
9   P04925 200533 27242144
10  P04925 200533 27242143
```

5   Case study

To illustrate how to use these features, "chicken" data package from bioconductor.org is utilized for demonstration. The following R code enable user to map data from Entrez Gene id to UniProtKB Accession number (in the package, 24 represents Entrez Gene id, and 19 represents UniProt Accession Number).

```
# load "chicken" data package from Bioconductor
>library(chicken)
>xx <- as.list(chickenENTREZID)

# collect multiple entries for one probe and delete "NA" entries.
>uxx<-unlist(xx)
>myList<-unique(as.matrix(uxx[!is.na(uxx)]))

# map the Entrez Gene id to UniProt Accession Number
```

```
>library(BioIDMapper)
>bio.convert(myList, 24, 19)->result
```

Also, the result can be linked directly to the related database website for more detailed information about that specific ID.

```
>bio.link(result[2,2],19)
```

In addition, the data analysis module offers services to summarize mapping results.

```
>bio.sum(result, myList, F)
```