Using the new Annotation Packages Answers to the Exercises

Marc Carlson

29 July 2008

1 Answers To Exercises

First lets prep the R session to answer these questions:

```
> library("hgu95av2.db")
> library("org.Hs.eg.db")
> library("org.Mm.eg.db")
> library("hom.Hs.inp.db")
```

EXERCISE 1: Query one of the other database tables.

```
> sql <- "SELECT * FROM omim LIMIT 10;"
> dbGetQuery(hgu95av2_dbconn(), sql)

   _id omim_id
1    4  108345
2    5  243400
3    7  107280
4    8  600338
5    9  603488
6   10  600950
7   11  601065
8   13  137150
9   14  143890
10  14  205400
```

EXERCISE 2: Use revmap to find out which probes map to the gene symbol ”HOXA9”.

```
> mget("HOXA9", revmap(hgu95av2SYMBOL))

$HOXA9
[1] "37809_at" "37810_at"
```

EXERCISE 3: How could you used mget() to learn all possible aliased gene symbols for pbids? And how could you use subset() to find out which of these probesets are="ADA"?

To use mget() to find pbids that match "ADA":

```
> pbids <- c("38912_at", "41654_at", "907_at", "2053_at", "2054_g_at",
+     "40781_at")
> mget(pbids, revmap(hgu95av2ALIAS2PROBE))

$`38912_at`
[1] "AAC2" "NAT2"

$`41654_at`
[1] "ADA"

$`907_at`
[1] "ADA"

$`2053_at`
[1] "CD325"  "CDHN"   "CDw325" "NCAD"   "CDH2"

$`2054_g_at`
[1] "CD325"  "CDHN"   "CDw325" "NCAD"   "CDH2"

$`40781_at`
[1] "DKFZP434N0250" "PKB-GAMMA"     "PKBG"          "PRKBG"
[5] "RAC-PK-gamma"  "RAC-gamma"     "STK-2"         "AKT3"
```

To use subset() to find which of the pbids above are = "ADA":

```
> regionalSubset <- subset(revmap(hgu95av2ALIAS2PROBE), Lkeys = pbids,
+     Rkeys = "ADA")
> toTable(regionalSubset)

  probe_id alias_symbol
1 41654_at          ADA
2   907_at          ADA
```

EXERCISE 4: Now that you have seen how to map a gene from human to mouse, try mapping the gene "Shh" back from mouse to man.

Do the whole thing in reverse.

```
> mget("Shh", org.Mm.egSYMBOL2EG)

$Shh
[1] "20423"
```

```
> mget("20423", org.Mm.egMGI)

$`20423`
[1] "MGI:98297"

> mget("MGI:98297", revmap(hom.Hs.inpMUSMU))

$`MGI:98297`
[1] "ENSP00000297261"

> mget("ENSP00000297261", org.Hs.egENSEMBLPROT2EG)

$ENSP00000297261
[1] "6469"

> mget("6469", org.Hs.egSYMBOL)

$`6469`
[1] "SHH"
```

EXERCISE 5: Query one of the tables and filter out entries that are not human and where the score = 1.

```
> sql <- "SELECT * FROM entamoeba_histolytica WHERE species='HOMSA' AND score!=1 LIMIT 10;"
> dbGetQuery(hom.Hs.inp_dbconn(), sql)

            inp_id clust_id species  score seed_status
1  ENSP00000317123        4   HOMSA 0.0694
2  ENSP00000263200        5   HOMSA 0.7613
3  ENSP00000269577        9   HOMSA   0.44
4  ENSP00000211402       13   HOMSA 0.9932
5  ENSP00000273353       15   HOMSA 0.1008
6  ENSP00000252172       15   HOMSA 0.1522
7  ENSP00000262873       15   HOMSA 0.1522
8  ENSP00000226207       15   HOMSA 0.1638
9  ENSP00000354976       15   HOMSA 0.1642
10 ENSP00000245503       15   HOMSA 0.1649
```

EXERCISE 6: Now use the fact that the clust_id is shared within this table between matching keys to try to recreate the data that is in the original inparanoid mapping.

1st we will need to get stuff from the human inparanoid database. Specifically, we will only want the stuff that is in the mus_musculus table.

```
> ## Look at the tables
> dbListTables(hom.Hs.inp_dbconn())
```

```
 [1] "aedes_aegypti"             "anopheles_gambiae"
 [3] "apis_mellifera"            "arabidopsis_thaliana"
 [5] "bos_taurus"                "caenorhabditis_briggsae"
 [7] "caenorhabditis_elegans"    "caenorhabditis_remanei"
 [9] "candida_glabrata"          "canis_familiaris"
[11] "ciona_intestinalis"        "cryptococcus_neoformans"
[13] "danio_rerio"               "debaryomyces_hanseneii"
[15] "dictyostelium_discoideum"  "drosophila_melanogaster"
[17] "drosophila_pseudoobscura"  "entamoeba_histolytica"
[19] "escherichia_coliK12"       "gallus_gallus"
[21] "gasterosteus_aculeatus"    "kluyveromyces_lactis"
[23] "macaca_mulatta"            "map_counts"
[25] "map_metadata"              "metadata"
[27] "monodelphis_domestica"     "mus_musculus"
[29] "oryza_sativa"              "pan_troglodytes"
[31] "rattus_norvegicus"         "saccharomyces_cerevisiae"
[33] "schizosaccharomyces_pombe" "sqlite_stat1"
[35] "takifugu_rubripes"         "tetraodon_nigroviridis"
[37] "xenopus_tropicalis"        "yarrowia_lipolytica"
```

```
> ## Look at the fields of "mus_musculus"
> dbListFields(hom.Hs.inp_dbconn(), "mus_musculus")
```

```
[1] "inp_id"      "clust_id"    "species"     "score"       "seed_status"
```

```
> ## Now lets look at some table contents to see what we have...
> dbGetQuery(hom.Hs.inp_dbconn(), "SELECT * FROM mus_musculus LIMIT 10;")
```

```
            inp_id clust_id species score seed_status
1  ENSP00000296619        1   HOMSA     1        100%
2      MGI:1274784        1   MUSMU     1        100%
3  ENSP00000352608        2   HOMSA     1        100%
4        MGI:99659        2   MUSMU     1        100%
5  ENSP00000243077        3   HOMSA     1        100%
6        MGI:96828        3   MUSMU     1        100%
7  ENSP00000355533        4   HOMSA     1        100%
8        MGI:99685        4   MUSMU     1        100%
9  ENSP00000261609        5   HOMSA     1        100%
10      MGI:103234        5   MUSMU     1        100%
```

```
> ## This will involve a quick "self join" of this table onto itself.
> ## Basically we want to take this:
> sql <- "SELECT * FROM mus_musculus WHERE species='MUSMU' AND score=1 LIMIT 10;"
> dbGetQuery(hom.Hs.inp_dbconn(), sql)
```

```
        inp_id clust_id species score seed_status
1  MGI:1274784        1   MUSMU     1        100%
```

```
2    MGI:99659        2   MUSMU     1        100%
3    MGI:96828        3   MUSMU     1        100%
4    MGI:99685        4   MUSMU     1        100%
5   MGI:103234        5   MUSMU     1        100%
6  MGI:2151136        6   MUSMU     1        100%
7  MGI:1276108        7   MUSMU     1        100%
8   MGI:103147        8   MUSMU     1        100%
9  MGI:2179432        9   MUSMU     1        100%
10  MGI:107714       10   MUSMU     1        100%
```

```
> ## And then we need to join that with this:
> sql <-  "SELECT * FROM mus_musculus WHERE species='HOMSA' AND score=1 LIMIT 10;"
> dbGetQuery(hom.Hs.inp_dbconn(), sql)
```

```
            inp_id clust_id species score seed_status
1  ENSP00000296619        1   HOMSA     1        100%
2  ENSP00000352608        2   HOMSA     1        100%
3  ENSP00000243077        3   HOMSA     1        100%
4  ENSP00000355533        4   HOMSA     1        100%
5  ENSP00000261609        5   HOMSA     1        100%
6  ENSP00000307314        6   HOMSA     1        100%
7  ENSP00000261359        7   HOMSA     1        100%
8  ENSP00000351750        8   HOMSA     1        100%
9  ENSP00000353197        9   HOMSA     1        100%
10 ENSP00000333363       10   HOMSA     1        100%
```

```
> ## Which will look something like this:
> sql <- "SELECT Mm.inp_id,Hs.inp_id FROM
+ (SELECT * FROM mus_musculus WHERE species='MUSMU' AND score=1) AS Mm,
+ (SELECT * FROM mus_musculus WHERE species='HOMSA' AND score=1) AS Hs
+ WHERE Mm.clust_id = Hs.clust_id;"
> HsMm <- dbGetQuery(hom.Hs.inp_dbconn(), sql)
> head(HsMm)
```

```
    Mm.inp_id       Hs.inp_id
1 MGI:1274784 ENSP00000296619
2   MGI:99659 ENSP00000352608
3   MGI:96828 ENSP00000243077
4   MGI:99685 ENSP00000355533
5  MGI:103234 ENSP00000261609
6 MGI:2151136 ENSP00000307314
```

EXERCISE 7: How would you change the above SQL query to only look for evidence types of ’IPI’ and ’IDA’?

```
> SQL <- paste("SELECT symbol FROM go_bp INNER JOIN gene_info USING(_id)",
+              "WHERE go_bp.evidence in ('IPI', 'IDA')")
```

```
> SQLans <- dbGetQuery(hgu95av2_dbconn(), SQL)
> SQLans <- unique(as.vector(t(SQLans)))
> length(SQLans)
```

```
[1] 1027
```

```
> SQLans[1:10]
```

```
 [1] "ABCA1" "ABCA2" "ABCA3" "ABL1"  "ABL2"  "ACO1"  "ACR"   "ACO2"  "ACTN4"
[10] "ACTN2"
```

EXERCISE 8: Can you join across a couple of databases? Make a join across the mapping in the hom.Hs.inp database to the ensemble_prot mapping in the org.Hs.eg.db database.

```
> ## First We will have to look at the data in the organism package:
> as.list(org.Hs.egENSEMBLPROT)[1:10]
```

```
$`1`
[1] "ENSP00000263100"

$`2`
[1] NA

$`3`
[1] NA

$`9`
[1] "ENSP00000307218"

$`10`
[1] "ENSP00000286479"

$`11`
[1] NA

$`12`
[1] "ENSP00000376795" "ENSP00000261981" "ENSP00000376793" "ENSP00000369712"

$`13`
[1] "ENSP00000232892"

$`14`
[1] "ENSP00000248450"

$`15`
[1] "ENSP00000250615" "ENSP00000376282"
```

```
> ## List the tables for what will be the local DB
> dbListTables(org.Hs.eg_dbconn())

 [1] "accessions"            "alias"                 "chrlengths"
 [4] "chromosome_locations"  "chromosomes"           "cytogenetic_locations"
 [7] "ec"                    "ensembl"               "ensembl_prot"
[10] "ensembl_trans"         "gene_info"             "genes"
[13] "go_bp"                 "go_bp_all"             "go_cc"
[16] "go_cc_all"             "go_mf"                 "go_mf_all"
[19] "kegg"                  "map_counts"            "map_metadata"
[22] "metadata"              "omim"                  "pfam"
[25] "prosite"               "pubmed"                "refseq"
[28] "sqlite_stat1"          "unigene"

> ## Look at the field for the table of interest
> dbListFields(org.Hs.eg_dbconn(), "ensembl_trans")

[1] "_id"      "trans_id"

> ## look at the data in the table of interest
> dbGetQuery(org.Hs.eg_dbconn(), "SELECT * FROM ensembl_prot LIMIT 10;")

   _id         prot_id
1    1 ENSP00000263100
2    4 ENSP00000307218
3    5 ENSP00000286479
4    7 ENSP00000376795
5    7 ENSP00000261981
6    7 ENSP00000376793
7    7 ENSP00000369712
8    8 ENSP00000232892
9    9 ENSP00000248450
10  10 ENSP00000250615

> ## look at that other table of interest:
> dbGetQuery(org.Hs.eg_dbconn(), "SELECT * FROM genes LIMIT 10;")

   _id gene_id
1    1       1
2    2       2
3    3       3
4    4       9
5    5      10
6    6      11
7    7      12
8    8      13
9    9      14
10  10      15
```

```
> ## We also have to combine these two queries like this:
> sql <- "SELECT e.prot_id,g.gene_id FROM
+ (SELECT * FROM ensembl_prot) AS e,
+ (SELECT * FROM genes) AS g
+ WHERE e._id = g._id;"
> EnsProt <- dbGetQuery(org.Hs.eg_dbconn(), sql)
> head(EnsProt)

        e.prot_id g.gene_id
1 ENSP00000263100         1
2 ENSP00000307218         9
3 ENSP00000286479        10
4 ENSP00000376795        12
5 ENSP00000261981        12
6 ENSP00000376793        12

> ## Now recall the results from the internal join we did before in exercise #6:
> sql <- "SELECT Mm.inp_id,Hs.inp_id FROM
+ (SELECT * FROM mus_musculus WHERE species='MUSMU' AND score=1) AS Mm,
+ (SELECT * FROM mus_musculus WHERE species='HOMSA' AND score=1) AS Hs
+ WHERE Mm.clust_id = Hs.clust_id;"
> HsMm <- dbGetQuery(hom.Hs.inp_dbconn(), sql)
> head(HsMm)

    Mm.inp_id       Hs.inp_id
1 MGI:1274784 ENSP00000296619
2   MGI:99659 ENSP00000352608
3   MGI:96828 ENSP00000243077
4   MGI:99685 ENSP00000355533
5  MGI:103234 ENSP00000261609
6 MGI:2151136 ENSP00000307314

> ## And remember that we have to attach this other database in order to join across
> homDBLoc <- system.file("extdata", "hom.Hs.inp.sqlite", package="hom.Hs.inp.db")
> attachSQL <- paste("ATTACH '", homDBLoc, "' as hDB;", sep = "")
> dbGetQuery(org.Hs.eg_dbconn(), attachSQL)

NULL

> ## And now we can assemble the final composite query
> sql <- "
+ SELECT * FROM
+  (SELECT e.prot_id as EID,g.gene_id FROM
+   (SELECT * FROM ensembl_prot) AS e,
+   (SELECT * FROM genes) AS g
+    WHERE e._id = g._id)                 AS ep,
+  (SELECT Mm.inp_id,Hs.inp_id as HID FROM
+   (SELECT * FROM hDB.mus_musculus WHERE species='MUSMU' AND score=1) AS Mm,
+   (SELECT * FROM hDB.mus_musculus WHERE species='HOMSA' AND score=1) AS Hs
+    WHERE Mm.clust_id = Hs.clust_id)   AS hom
+ WHERE ep.EID = hom.HID;"
> HugeJoin <- dbGetQuery(org.Hs.eg_dbconn(), sql)
> head(HugeJoin)
```

```
              EID g.gene_id   Mm.inp_id             HID
1 ENSP00000352608      6261   MGI:99659 ENSP00000352608
2 ENSP00000243077      4035   MGI:96828 ENSP00000243077
3 ENSP00000261609      8924  MGI:103234 ENSP00000261609
4 ENSP00000261359     57448 MGI:1276108 ENSP00000261359
5 ENSP00000353197     23077 MGI:2179432 ENSP00000353197
6 ENSP00000333363      1769  MGI:107714 ENSP00000333363
```

```
> ## Don't forget to clean up:
> detachSQL <- paste("DETACH hDB")
> dbGetQuery(org.Hs.eg_dbconn(), detachSQL)
```

```
NULL
```