[Ortholog] Mapping the Applied Biosystems Human/Mouse Survey v2.0 Micro-arrays to Ensembl Gene Identifiers

Werner Van Belle
e-mail: werner.van.belle@gmail.com, werner@onlinux.be
Medical Genetics (MEDGEN)
University Hospital Noprthern Norway (UNN)
Tromsø; Norway

June 4, 2007

Abstract

This document describes cross joining of gene tables between Celera's mouse genome identifiers, Celera human genome identifiers and the more useful Ensembl identifiers. The context in which this research is set are the genes FKRP and TAF4. By using siRNA's we interfered with the transcription and measured their effect upon the transcriptome. The Applied Biosystems 1700 micro-array scanner measured and reported the transcription quantities. Two micro-array types were used: the human genome survey v2.0 micro-arrays and the mouse genome survey v2.0 micro-arrays. Based on the different micro-array measurements we wanted to predict which proteins would be influenced in a cell system if we know the up/down regulation of the measured probes. To this end we wanted to use the human protein interaction map (as defines by Rual'06), which uses Ensembl annotated genes. This of course formed a major problem. First, the Applied Biosystems scanner does not export Ensembl gene annotations. Secondly, the human protein interaction map might not be a good model for a mouse micro-array, so we needed to go through various orthologs. This document tells two stories: first, and most annoyingly: how to get Ensembl identifiers into an Applied Biosystems micro-array. Secondly, and slightly more interesting, how to retrieve a mouse to human ortholog mapping from Ensembl.

Contents

1 Convention

In this document we work with three forms of information. First there is a local database, which we call FkrpTaf4. It will contain all the information we need to perform further analysis. Secondly there is the public accessible ensembldb. Most people know this through the web-interface. What is probably lesser known is that this database is also immediately accessible through SQL, which makes it optimal for our purposes (http://ensembldb.ensembl.org). The third set of information are local comma separated file (CSV), which are used a) to transfer information from the Ensembl database into our own FkrpTaf4 database and b) to import data from the Applied Biosystems into our FkrpTaf4 database).

For each query we do not mention whether something is a temporary table neither do we drop tables if they already exist. This is properly done in the actual SQL files, but since it is of little relevance to understanding what is happening we omit this information here. There is however one catch: if a table is a temporary table it cannot be reused within the same query. In that case, one might need to make a copy into a second temporary table. For each created table we also list a small example output that illustrates the contents of each table.

To execute a query on a database one can do the following `mysql --user=werner -D FkrpTaf4 -A --batch <import-ab.sql`

For the Ensembl database one can use `mysql --user=anonymous -h ensembldb.ensembl.org -D homo_sapiens_core_44_36f -A --batch <swissprot2ensembl.sql >imports/swissprot2ensg.csv` or `mysql --user=anonymous -h ensembldb.ensembl.org -D mus_musculus_core_44_36e -A --batch <swissprot2ensembl.sql >imports/swissprot2ensmusg.csv`

2 Importing Applied Biosystems Tables

We measured the influence of FKRP and TAF4 on various cell systems through measuring the transcriptional changes with human and mouse genome survey arrays. Once this was done, we aimed to integrate this data into a protein interaction map as to find the proteins that are likely influenced mostly by the proteins of interest. We encountered some major obstacles to this approach. First, the Applied Biosystems 1700 scanner does not provide Ensembl annotated genes. Secondly, the Applied Biosystems 1700 scanner does not provide the probe sequences, making an automatic mapping to either the mouse or human genome more complicated than it should be and thirdly: exporting the Unigene/Swissprot annotated genes from the Applied Biosystems 1700 machine was prohibitively slow. In the end we exported all tables using a tedious 10 columns at a time approach. This lead to 6 tables that we could join afterward. One conducting high throughput proteomics will find this 'small bug' a major issue, since the export times become prohibitively long. In other words: aside from the fact that Applied Biosystems provides only 56% useful measurements, the Applied Biosystems 1700 scanner also seems rather unusable in high throughput settings.

2.1 Loading

To import the Applied Biosystems tables, we exported three different experiments in 6 separate files. We also had to clean out some spaces that were added in the allset1 data. When this operation is performed, we have three tables: FkrpSiRna1, FkrpSiRna2 and FkrpScrambled. The file `import-ab` contains all the details on the import process.

2.1.1 Set 1

To illustrate the mechanism, we elaborate somewhat on allset1. The target table must first be created, thereby reflecting the columns as they occur in the original Applied Biosystems CSV tables. We also introduce assay_name, probe_id, gene_name and sample_name as indices since we later need to join on these columns. If we don't do this, most operations will be extremely slow. The query below is ran on the FkrpTaf4 schema.

```
CREATE TABLE allset1
  (Assay_Name VARCHAR(128),
  INDEX (Assay_Name, Probe_ID, Gene_ID, Sample_Name),
  Row FLOAT,
  Col FLOAT,
  Probe_ID VARCHAR(128),
  Probe_Type TEXT,
  Gene_ID VARCHAR(128),
  X FLOAT,
  Y FLOAT,
  Assay_Normalized_Signal FLOAT,
  Signal FLOAT,
  CL_Sig FLOAT,
  CL_Raw FLOAT,
  SDEV FLOAT,
  CV FLOAT,
  S_N FLOAT,
  CL_Sig_Error FLOAT,
  CL_Raw_SDEV FLOAT,
  Flags INT,
  Sample_Name VARCHAR(128),
  id INT AUTO_INCREMENT PRIMARY KEY);
```

Once the table is created we can import the data with:

```
LOAD DATA LOCAL INFILE 'imports/all-set1.ab.csv'
INTO TABLE allset1;
```

The query above is ran on the FkrpTaf4 schema.

Cleaning up Of course, it would be nice if the sample_name could be compared between table1 and table6. In practice, this could not be done since alslet1 included a 10 character at the end of each sample name. To get rid of those we needed the following update. The query below is ran on the FkrpTaf4 schema.

```
REPLACE allset1 SELECT
  Assay_name,row,col,probe_id,probe_type,
  gene_id,x,y,assay_normalized_signal,
  signal,cl_sig,cl_raw,sdev,cv,s_n,
  cl_sig_error,cl_raw _sdev,flags,
  Trim('\r' FROM sample_name),id
FROM allset1;
```

This gives a table consisting of

```
+-------------------------+------+------+----------+------------+--------------+
| Assay_name              | Row  | Col  | Probe_id | Probe_type | gene_id      |
+-------------------------+------+------+----------+------------+--------------+
| HB00588 3/1/07 12:12 PM |  189 |   77 | 100002   | probe      | hCG1643199.4 |
| HB00588 3/1/07 12:12 PM |   45 |   39 | 100003   | probe      | hCG2041918   |
| HB00588 3/1/07 12:12 PM |  109 |  152 | 100027   | probe      | hCG31426.2   |
| HB00588 3/1/07 12:12 PM |   70 |    7 | 100036   | probe      | hCG1979099.1 |
| HB00588 3/1/07 12:12 PM |  173 |  129 | 100037   | probe      | hCG42687.4   |
| HB00588 3/1/07 12:12 PM |  114 |   54 | 100039   | probe      | hCG2015782   |
| HB00588 3/1/07 12:12 PM |   68 |   51 | 100044   | probe      | hCG36953.3   |
| HB00588 3/1/07 12:12 PM |  153 |  123 | 100045   | probe      | hCG1776836.3 |
| HB00588 3/1/07 12:12 PM |   75 |  157 | 100051   | probe      | hCG1642464.3 |
| HB00588 3/1/07 12:12 PM |  146 |  139 | 100052   | probe      | hCG22993.3   |
+---------+---------+-------------------------+---------+---------+---------+---------+
| x       | y       | assay_normalized_signal | signal  | cl_sig  | cl_raw  | sdev    |
+---------+---------+-------------------------+---------+---------+---------+---------+
|  991.21 | 1135.54 |                  161.22 |  171638 |  170314 |  177617 | 1233.32 |
|  575.93 |  875.26 |                    0.15 |  194.56 |  140.64 |  3674.4 |  194.56 |
| 1792.77 | 1560.84 |                     0.2 |  251.37 |  -38.76 | 4600.85 |  251.37 |
|  232.78 | 1145.18 |                    8.32 | 10512.5 | 11065.4 | 16749.1 |  691.04 |
| 1549.67 |  960.67 |                   44.04 | 46886.9 | 45509.2 | 51538.9 |  590.25 |
|   740.1 | 1617.52 |                   11.22 | 14174.1 | 13962.3 | 20199.8 |  579.15 |
|  706.61 | 1122.22 |                    0.17 |  212.63 |   -75.6 |  4337.4 |  212.63 |
| 1484.97 |  746.02 |                    0.26 |  277.94 | -626.52 | 4945.81 |  277.94 |
| 1846.74 | 1194.41 |                    0.23 |  291.73 | -496.82 | 4405.79 |  291.73 |
| 1655.39 |  670.39 |                    0.58 |  621.79 |  -64.09 | 6495.53 |  621.79 |
+-------+--------+--------------+-------------+-------+-------------+----+
| cv    | s_n    | CL_sig_error | CL_Raw_sdev | Flags | sample_name | id |
+-------+--------+--------------+-------------+-------+-------------+----+
|  0.05 | 139.17 |       296.57 |           0 |     0 | II-1        |  2 |
|  1.14 |   0.88 |       103.51 |           0 |     1 | II-1        |  3 |
| 31.09 |  -0.03 |       131.81 |           0 |     1 | II-1        |  4 |
|  0.08 |  15.21 |       173.15 |           0 |     0 | II-1        |  5 |
|  0.05 |  79.44 |        174.7 |           0 |     0 | II-1        |  6 |
|  0.06 |  24.47 |       189.76 |           0 |     0 | II-1        |  7 |
|  4.14 |  -0.24 |       125.93 |           0 |     1 | II-1        |  8 |
|  0.51 |  -1.96 |        98.86 |           0 |     1 | II-1        |  9 |
|  0.61 |  -1.65 |       138.64 |           0 |     1 | II-1        | 10 |
|  5.63 |  -0.18 |       113.36 |           0 |     1 | II-1        | 11 |
```

2.1.2 Set 6

The following query imports, among other things, the important mCG, Swissprot and Unigene identifiers into the FkrpTaf4 database. The query below is ran on the FkrpTaf4 schema.

```
CREATE TABLE allset6
  (Assay_Name VARCHAR(128),
  Probe_ID VARCHAR(128),
  Gene_ID VARCHAR(128),
  Sample_Name VARCHAR(128),
  INDEX (Assay_Name, Probe_ID, Gene_ID, Sample_Name),
  Status TEXT,
  SwissProt TEXT,
  UniGene TEXT,
  dbEST TEXT,
  hCG TEXT,
  hCP TEXT,
  hCT TEXT,
  mCG TEXT,
  mCP TEXT,
  mCT TEXT,
  rCG TEXT,
  rCP TEXT,
  rCT TEXT);
LOAD DATA LOCAL INFILE 'imports/all-set6.ab.csv'
INTO TABLE allset6;
```

This gives a table as:

```
+-------------------------+----------+--------------+-------------+------------+
| assay_name              | probe_id | gene_id      | sample_name | status     |
+-------------------------+----------+--------------+-------------+------------+
| HB00588 3/1/07 12:12 PM | 100002   | hCG1643199.4 | II-1        | pseudogene |
| HB00588 3/1/07 12:12 PM | 100003   | hCG2041918   | II-1        | current    |
| HB00588 3/1/07 12:12 PM | 100027   | hCG31426.2   | II-1        | current    |
| HB00588 3/1/07 12:12 PM | 100036   | hCG1979099.1 | II-1        | current    |
| HB00588 3/1/07 12:12 PM | 100037   | hCG42687.4   | II-1        | current    |
| HB00588 3/1/07 12:12 PM | 100039   | hCG2015782   | II-1        | current    |
| HB00588 3/1/07 12:12 PM | 100044   | hCG36953.3   | II-1        | current    |
| HB00588 3/1/07 12:12 PM | 100045   | hCG1776836.3 | II-1        | current    |
| HB00588 3/1/07 12:12 PM | 100051   | hCG1642464.3 | II-1        | current    |
| HB00588 3/1/07 12:12 PM | 100052   | hCG22993.3   | II-1        | current    |
+------------------------------------+-----------+---------------------------------------------------+
| swissprot                          | unigene   | hcg                                               |
+------------------------------------+-----------+---------------------------------------------------+
|                                    |           | hCG1643199.4                                      |
|                                    |           | hCG2041918                                        |
| O95201;P13682;P17027;P51523;Q15776 | Hs.57679  | hCG31426.2                                        |
| Q92610                             | Hs.368756 | hCG1820838.1;hCG1979099.1;hCG1989348;hCG1994281.1 |
|                                    | Hs.302903 | hCG42687.4                                        |
| Q9UJX3                             | Hs.530379 | hCG2015782                                        |
|                                    | Hs.278954 | hCG36953.3                                        |
|                                    |           | hCG1776836.3                                      |
|                                    |           | hCG1642464.3                                      |
|                                    | Hs.199068 | hCG22993.3                                        |
```

2.2 Extracting FKRP related tables

We extract 3 different tables: FkrpSiRna1, FkrpSiRna2 and FkrpScrambled using the following SQL statements. Each table will have duplicate rows for specific genes. This is because they have also been measured multiple times. FkrpSiRna2 is smaller than the two others since we only had two slides and not three.

2.2.1 Creating FkrpSiRNA1

Allset1 and allset6 are imported in 2.1.1 and 2.1.2. The query below is ran on the FkrpTaf4 schema.

```
CREATE TABLE FkrpSiRna1
SELECT allset1.Gene_ID, Assay_Normalized_Signal
FROM allset6, allset1
WHERE (allset6.Sample_Name="1-1"
    or allset6.Sample_Name="2-1"
    or allset6.Sample_Name="3-1")
  and allset6.sample_name=allset1.sample_name
  and allset1.Assay_Name=allset6.Assay_Name
  and allset1.gene_id=allset6.gene_id
  and allset1.Probe_ID=allset6.Probe_ID;
```

Table example:

```
+--------------+-------------------------+
| Gene_ID      | Assay_Normalized_Signal |
+--------------+-------------------------+
| AK079773.1   |                    1.58 |
| mCG9222.3    |                   19.46 |
| mCG5316.2    |                    0.31 |
| mCG1036527.1 |                    6.28 |
| mCG1050139   |                     0.2 |
| mCG121612    |                   36.77 |
| mCG142727    |                    0.14 |
| mCG130331.1  |                    0.35 |
| mCG1045481.1 |                    1.99 |
| mCG141353    |                    0.34 |
```

2.2.2 Creating FkrpSiRNA2

Allset1 and allset6 are imported in 2.1.1 and 2.1.2. The query below is ran on the FkrpTaf4 schema.

```
CREATE TABLE FkrpSiRna2
SELECT allset1.Gene_ID, Assay_Normalized_Signal
FROM allset6, allset1
WHERE (allset6.Sample_Name="1-2"
or allset6.Sample_Name="2-2")
and allset6.sample_name=allset1.sample_name
and allset1.Assay_Name=allset6.Assay_Name
and allset1.gene_id=allset6.gene_id
and allset1.Probe_ID=allset6.Probe_ID;
+--------------+-------------------------+
| Gene_ID      | Assay_Normalized_Signal |
+--------------+-------------------------+
| AK079773.1   |                    1.33 |
| mCG9222.3    |                   15.48 |
| mCG5316.2    |                    0.19 |
| mCG1036527.1 |                       4 |
| mCG1050139   |                     0.2 |
| mCG121612    |                    40.8 |
| mCG142727    |                    0.21 |
| mCG130331.1  |                    0.45 |
| mCG1045481.1 |                     6.6 |
| mCG141353    |                    0.33 |
```

2.2.3 Creating FkrpScrambled

Allset1 and allset6 are imported in 2.1.1 and 2.1.2. The query below is ran on the FkrpTaf4 schema.

```
CREATE TABLE FkrpScrambled
SELECT allset1.Gene_ID, Assay_Normalized_Signal
FROM allset6, allset1
```

```
WHERE (allset6.Sample_Name="1-3"
    or allset6.Sample_Name="2-3"
    or allset6.Sample_Name="3-3")
  and allset6.sample_name=allset1.sample_name
  and allset1.Assay_Name=allset6.Assay_Name
  and allset1.gene_id=allset6.gene_id
  and allset1.Probe_ID=allset6.Probe_ID;
```

Example

```
+--------------+-------------------------+
| Gene_ID      | Assay_Normalized_Signal |
+--------------+-------------------------+
| AK079773.1   |                    1.67 |
| mCG9222.3    |                   14.78 |
| mCG5316.2    |                    0.96 |
| mCG1036527.1 |                    2.03 |
| mCG1050139   |                    0.12 |
| mCG121612    |                   40.13 |
| mCG142727    |                    0.13 |
| mCG130331.1  |                    0.26 |
| mCG1045481.1 |                    1.93 |
| mCG141353    |                    0.23 |
```

2.3 Extracting TAF4 related tables

2.3.1 Creating Taf4SiRnaHela

The query below is ran on the FkrpTaf4 schema.

```
CREATE TABLE Taf4SiRnaHela
SELECT allset1.Gene_ID, Assay_Normalized_Signal
FROM allset6, allset1
WHERE (allset6.Sample_Name="I-1"
or allset6.Sample_Name="I-2"
or allset6.Sample_Name="I-3"
or allset6.Sample_Name="I-4")
and allset6.sample_name=allset1.sample_name
and allset1.Assay_Name=allset6.Assay_Name
and allset1.gene_id=allset6.gene_id
and allset1.Probe_ID=allset6.Probe_ID;
```

Example

```
+--------------+-------------------------+
| Gene_ID      | Assay_Normalized_Signal |
+--------------+-------------------------+
| hCG2041918   |                    0.08 |
| hCG31426.2   |                    0.39 |
| hCG1979099.1 |                    7.13 |
| hCG42687.4   |                   40.41 |
| hCG2015782   |                    9.79 |
| hCG36953.3   |                    0.08 |
| hCG1776836.3 |                    0.37 |
| hCG1642464.3 |                    0.21 |
| hCG22993.3   |                    0.17 |
| hCG1793655.1 |                    3.59 |
```

2.3.2 Creating Taf4ScrambledHela

The query below is ran on the FkrpTaf4 schema.

```
CREATE TABLE Taf4ScrambledHela
SELECT allset1.Gene_ID, Assay_Normalized_Signal
FROM allset6, allset1
WHERE (allset6.Sample_Name="II-1"
or allset6.Sample_Name="II-2"
or allset6.Sample_Name="II-3")
and allset6.sample_name=allset1.sample_name
and allset1.Assay_Name=allset6.Assay_Name
and allset1.gene_id=allset6.gene_id
and allset1.Probe_ID=allset6.Probe_ID;
```

Example

```
+--------------+-------------------------+
| Gene_ID      | Assay_Normalized_Signal |
+--------------+-------------------------+
| hCG2041918   |                    0.15 |
| hCG31426.2   |                     0.2 |
| hCG1979099.1 |                    8.32 |
| hCG42687.4   |                   44.04 |
| hCG2015782   |                   11.22 |
| hCG36953.3   |                    0.17 |
| hCG1776836.3 |                    0.26 |
| hCG1642464.3 |                    0.23 |
| hCG22993.3   |                    0.58 |
| hCG1793655.1 |                     4.2 |
```

2.3.3 Creating Taf4SiRnaSkndz

The query below is ran on the FkrpTaf4 schema.

```
CREATE TABLE Taf4SiRnaSkndz
SELECT allset1.Gene_ID, Assay_Normalized_Signal
FROM allset6, allset1
WHERE (allset6.Sample_Name="Si I"
or allset6.Sample_Name="Si II"
or allset6.Sample_Name="Si III"
or allset6.Sample_Name="Si IV")
and allset6.sample_name=allset1.sample_name
and allset1.Assay_Name=allset6.Assay_Name
and allset1.gene_id=allset6.gene_id
and allset1.Probe_ID=allset6.Probe_ID;
```

Example

```
+--------------+-------------------------+
| Gene_ID      | Assay_Normalized_Signal |
+--------------+-------------------------+
| hCG2041918   |                    0.16 |
| hCG31426.2   |                    0.12 |
| hCG1979099.1 |                    8.83 |
| hCG42687.4   |                   53.12 |
| hCG2015782   |                   14.13 |
| hCG36953.3   |                    0.11 |
| hCG1776836.3 |                    0.22 |
| hCG1642464.3 |                    0.14 |
| hCG22993.3   |                    0.54 |
| hCG1793655.1 |                    1.02 |
```

2.3.4 Creating Taf4ScrambledSkndz

The query below is ran on the FkrpTaf4 schema.

```
CREATE TABLE Taf4ScrambledSkndz
SELECT allset1.Gene_ID, Assay_Normalized_Signal as signal
FROM allset6, allset1
WHERE (allset6.Sample_Name="Scr I"
or allset6.Sample_Name="Scr II"
or allset6.Sample_Name="Scr III"
or allset6.Sample_Name="Scr IV")
and allset6.sample_name=allset1.sample_name
and allset1.Assay_Name=allset6.Assay_Name
and allset1.gene_id=allset6.gene_id
and allset1.Probe_ID=allset6.Probe_ID;
```

Example

```
+--------------+-------------------------+
| Gene_ID      | Assay_Normalized_Signal |
+--------------+-------------------------+
| hCG2041918   |                    0.55 |
| hCG31426.2   |                    0.16 |
| hCG1979099.1 |                   10.38 |
| hCG42687.4   |                   56.67 |
| hCG2015782   |                   11.33 |
| hCG36953.3   |                     0.3 |
| hCG1776836.3 |                    0.31 |
| hCG1642464.3 |                    1.23 |
| hCG22993.3   |                    0.32 |
| hCG1793655.1 |                    1.61 |
```

3 Mapping mCG/hCG gene identifiers to Ensembl gene identifiers

To integrate the human protein interaction map into the previously mentioned network, we need to go through the external annotated identifiers. The Applied Biosystems 1700 scanner can produce tables that include the Unigene and Swissprot identifiers. Based on these we can find the Ensembl genes and thus link them back to our hCG identifiers.

3.1 Ensembl database mappings

To make this work we first need to find various sources of data in the Ensembl database.

1. The basic schema/database of interest is homo_sapiens_core_44_36f
2. All genes in the Ensembl database have a unique internal number (the gene primary key), which does not match the ENSG annotation. These can be mapped to each other using the gene_stable_id table.
3. The linkage between a gene_id and an external database is given in the gene table where gene_id maps to a xref_id
4. The actual external database identifier can be found in the dbprimary_acc key in the xref table. For Swissprot the external_db_id should be 2200

3.1.1 Ensembl gene descriptions

The query below can be ran on the homo_sapiens_core_44_36f ensembldb schema or on the mus_musculus_core_44_36 schema. Depending on the choice of database we map ENSG... or ENSMUSG... identifiers to their description.

```
SELECT DISTINCT stable_id, description
FROM gene_stable_id JOIN gene using (gene_id);
```

Executed on the homo sapiens database

```
+-----------------+--------------------------------------------------------------------------------------
| stable_id       | description
+-----------------+--------------------------------------------------------------------------------------
| ENSG00000129824 | 40S ribosomal protein S4, Y isoform 1. [Source:Uniprot/SWISSPROT;Acc:P22090]
| ENSG00000067646 | Zinc finger Y-chromosomal protein. [Source:Uniprot/SWISSPROT;Acc:P08048]
| ENSG00000176679 | Homeobox protein TGIF2LY (TGFB-induced factor 2-like protein, Y- linked) (TGF(beta)indu
| ENSG00000168757 | testis specific protein, Y-linked 2 [Source:RefSeq_peptide;Acc:NP_072095]
| ENSG00000186406 | RNA binding motif (Fragment). [Source:Uniprot/SPTREMBL;Acc:Q13381]
| ENSG00000129816 | testis-specific transcript, Y-linked 1 (TTTY1) on chromosome Y [Source:RefSeq_dna;Acc:
| ENSG00000197285 | testis-specific transcript, Y-linked 2 (TTTY2) on chromosome Y [Source:RefSeq_dna;Acc:
| ENSG00000206198 | testis-specific transcript, Y-linked 21 (TTTY21) on chromosome Y [Source:RefSeq_dna;Ac
```

Executed on the mus_musculus database it returns

```
+--------------------+------------------------------------------------------------------------------------
| stable_id          | description
+--------------------+------------------------------------------------------------------------------------
| ENSMUSG00000053211 | zinc finger protein 2, Y linked [Source:MarkerSymbol;Acc:MGI:99213]
| ENSMUSG00000068457 | ubiquitously transcribed tetratricopeptide repeat gene, Y chromosome [Source:Marker
| ENSMUSG00000069053 | Ubiquitin-activating enzyme E1 Y (Ubiquitin-activating enzyme E1). [Source:Uniprot/
| ENSMUSG00000056673 | jumonji, AT rich interactive domain 1D (Rbp2 like) [Source:MarkerSymbol;Acc:MGI:9978
| ENSMUSG00000069049 | eukaryotic translation initiation factor 2, subunit 3, structural gene Y-linked [Sou
| ENSMUSG00000069045 | DEAD (Asp-Glu-Ala-Asp) box polypeptide 3, Y-linked [Source:MarkerSymbol;Acc:MGI:134
| ENSMUSG00000069044 | ubiquitin specific peptidase 9, Y chromosome [Source:MarkerSymbol;Acc:MGI:1313274]
| ENSMUSG00000069618 | RIKEN cDNA 1700012B15 gene [Source:MarkerSymbol;Acc:MGI:1921423]
| ENSMUSG00000020671 | RAB10, member RAS oncogene family [Source:MarkerSymbol;Acc:MGI:105066]
| ENSMUSG00000075505 | NULL
```

The results of this query will later on be imported into the FkrpTaf4 database (5.5) and then used in the final join (5.6).

3.1.2 Swissprot

To create a mapping from a Swissprot identifier to an Ensembl identifier requires us to join the stable_gene_id, gene and xref tables. In addition, it seems that sometimes multiple mappings are necessary. Namely, an external identifier can refer to a transcript *or* to the gene immediately. To resolve this we need the union of 2 queries.

Immediate gene mapping This is a query to map the external id immediately onto the gene. The query below can be ran on the homo_sapiens_core_44_36f ensembldb schema or on the mus_musculus_core_44_36e schema.

```
SELECT DISTINCT dbprimary_acc, stable_id, gene.description
FROM xref, gene, gene_stable_id
WHERE external_db_id=2200
and xref_id=display_xref_id
and gene.gene_id=gene_stable_id.gene_id;
```

Executed on the mus_musculus database it gives

```
+---------------+--------------------+---------------------------------------------------------------------
| dbprimary_acc | stable_id          | description
+---------------+--------------------+---------------------------------------------------------------------
| P46425        | ENSMUSG00000038155 | Glutathione S-transferase P 2 (EC 2.5.1.18) (GST YF-YF) (GST-piA) (
| Q8K2L9        | ENSMUSG00000033450 | T-cell activation Rho GTPase-activating protein (T-cell activation
| P83882        | ENSMUSG00000049751 | 60S ribosomal protein L36a (60S ribosomal protein L44). [Source:Uni
| P05531        | ENSMUSG00000054626 | X-linked lymphocyte-regulated protein PM1. [Source:Uniprot/SWISSPRO
| Q9WV98        | ENSMUSG00000021079 | Mitochondrial import inner membrane translocase subunit Tim9. [Sour
| Q6IE32        | ENSMUSG00000060201 | Serine protease inhibitor Kazal-type 7 precursor (Esophagus cancer-
| O88574        | ENSMUSG00000031609 | Histone deacetylase complex subunit SAP30 (Sin3-associated polypepti
| Q9JI46        | ENSMUSG00000024213 | Diphosphoinositol polyphosphate phosphohydrolase 1 (EC 3.6.1.52) (D
| Q03740        | ENSMUSG00000070870 | Gamma crystallin E. [Source:Uniprot/SWISSPROT;Acc:Q03740]
| P27545        | ENSMUSG00000055694 | LAG1 longevity assurance homolog 1 (UOG-1 protein). [Source:Uniprot/
```

External id to transcript to gene This query maps the external id to its transcript, which is then mapped onto its producing gene. The query below can be ran on the homo_sapiens_core_44_36f ensembldb schema or on the mus_musculus_core_44_36e schema.

```
SELECT dbprimary_acc, stable_id, gene.description
FROM xref, transcript, gene, gene_stable_id
WHERE external_db_id=2200
and xref_id=transcript.display_xref_id
and gene.gene_id=transcript.gene_id
and gene.gene_id=gene_stable_id.gene_id;
```

Executed on the mus_musculus database it gives slightly different results than the above query, illustrating that some genes are not immediately accessible through the straightforward mapping:

```
+---------------+--------------------+----------------------------------------------------------------------
| dbprimary_acc | stable_id          | description
+---------------+--------------------+----------------------------------------------------------------------
| P46425        | ENSMUSG00000038155 | Glutathione S-transferase P 2 (EC 2.5.1.18) (GST YF-YF) (GST-piA) (
| Q8BKE5        | ENSMUSG00000066307 | RIKEN cDNA E130016E03 gene [Source:MarkerSymbol;Acc:MGI:2444973]
| Q8K2L9        | ENSMUSG00000033450 | T-cell activation Rho GTPase-activating protein (T-cell activation 
| P83882        | ENSMUSG00000049751 | 60S ribosomal protein L36a (60S ribosomal protein L44). [Source:Uni
| P05531        | ENSMUSG00000054626 | X-linked lymphocyte-regulated protein PM1. [Source:Uniprot/SWISSPRO
| Q9WV98        | ENSMUSG00000021079 | Mitochondrial import inner membrane translocase subunit Tim9. [Sour
| Q9WV98        | ENSMUSG00000021079 | Mitochondrial import inner membrane translocase subunit Tim9. [Sour
| Q9WV98        | ENSMUSG00000021079 | Mitochondrial import inner membrane translocase subunit Tim9. [Sour
| Q6IE32        | ENSMUSG00000060201 | Serine protease inhibitor Kazal-type 7 precursor (Esophagus cancer-
| Q80WG5        | ENSMUSG00000007476 | phytanoyl-CoA dioxygenase domain containing 1 [Source:MarkerSymbol;
```

The merge of the two above queries then leads to

```
SELECT DISTINCT
  dbprimary_acc as swissprot,
  stable_id as ensembl,
  gene.description as description
FROM xref, gene, gene_stable_id
WHERE external_db_id=2200
AND xref_id=display_xref_id
AND gene.gene_id=gene_stable_id.gene_id
UNION DISTINCT
SELECT DISTINCT
  dbprimary_acc as swissprot,
  stable_id as ensembl,
  gene.description as description
FROM xref, transcript, gene, gene_stable_id
WHERE external_db_id=2200
AND xref_id=transcript.display_xref_id
AND gene.gene_id=transcript.gene_id
AND gene.gene_id=gene_stable_id.gene_id
```

If we execute this query on the Ensembl mus_musculus_core_44_36e database we obtain a mapping from swissprot2ensmusg. We store the results of this query in the table Swissprot2Mid as well, which will later on be used to create a shortened homologlist.

3.1.3 Unigene identifiers

Unigene identifiers are mapped as well through their transcripts but in a slightly different manner. By executing the following statement on the Ensembl database homo_sapiens_core_44_36f we obtain the unigene2ensg mapping (All stable_id's will be of the form ENSG000...). If we execute the query on the mus_musculus_core_44_36e database we obtain the unigene2ensmusg mapping (all id's will be of the form ENSMUSG000...).

```
SELECT DISTINCT
  dbprimary_acc as unigene,
  stable_id as ensembl
FROM xref, object_xref, transcript, gene_stable_id
WHERE external_db_id=4100
and object_xref.xref_id=xref.xref_id
and transcript.transcript_id=object_xref.ensembl_id
and gene_stable_id.gene_id=transcript.gene_id;
```

Example

```
+-----------+--------------------+
| unigene   | ensembl            |
+-----------+--------------------+
| Mm.27038  | ENSMUSG00000036083 |
| Mm.39752  | ENSMUSG00000021573 |
| Mm.41636  | ENSMUSG00000002733 |
| Mm.76494  | ENSMUSG00000028528 |
| Mm.253378 | ENSMUSG00000043556 |
| Mm.260194 | ENSMUSG00000020474 |
| Mm.268582 | ENSMUSG00000071256 |
| Mm.317248 | ENSMUSG00000066443 |
| Mm.317248 | ENSMUSG00000041453 |
| Mm.390885 | ENSMUSG00000014077 |
```

The table defined by the above query ion the mus_musculus schema will be called Unigene2Mid.

3.2 Mapping mCG to ensmusg identifiers

One might assume now that we can simply link the mCG identifiers through their Swissprot or Unigene link to the Ensembl human gene using one of the above tables. That is however incorrect. Such a table will be empty since no Swissprot identifier nor Unigene identifier from the human genome is reused in the mouse genome. Instead we need to go through an ortholog mapping. We start out with creating a mCG2ensmusg table first. This is of course again more tricky than it looks at first sight. The problem that we have now is that the list of Swissprot identifiers linked to each mCG gene is separated with semicolons (;). To find them we must thus work the other way around: create a set of all the Swissprot fuckers we have and find them back inside the mCG tables. To reduce calculation time we first create a small table containing all the mCG|Swissprot links

3.2.1 Identifier blobs

By appending a ';' to the end of each Swissprot identifier list, we are sure that every one of our Swissprot identifiers will be found when we attach a ; to it as well. If we don't do this we might find MM123 back in the list 'MM1234; MM44'. All that is left now is to make the joins of the swissport2ensmusg vs mcg2ensmusg and unigen2ensmusg vs mcg2ensmusg. The query below is ran in the FkrpTaf4 database.

```
CREATE TABLE CG2Blurb
SELECT DISTINCT
  Gene_ID,
  Concat(Trim(SwissProt),';') as swissprot,
  Concat(Trim(UniGene),';') as unigene
FROM allset6;
```

Example

```
+--------------+-------------------------------------+------------+
| Gene_ID      | swissprot                           | unigene    |
+--------------+-------------------------------------+------------+
| hCG31426.2   | O95201;P13682;P17027;P51523;Q15776; | Hs.57679;  |
| hCG1979099.1 | Q92610;                             | Hs.368756; |
| hCG42687.4   | ;                                   | Hs.302903; |
| hCG2015782   | Q9UJX3;                             | Hs.530379; |
| hCG36953.3   | ;                                   | Hs.278954; |
| hCG22993.3   | ;                                   | Hs.199068; |
| hCG33215.3   | ;                                   | Hs.30011;  |
| hCG22998.3   | ;                                   | Hs.294009; |
| hCG2039675   | Q8IUX4;Q9UH17;                      | Hs.337667; |
| hCG14966.2   | O14944;                             | Hs.115263; |
```

The mCG entries are taken from the 6th Applied Biosystems file (2.1.2).

3.2.2 Creating the mCG2Ensmusg table

This table creates a mapping from the various mCG identifiers we found to their associated ensmusg identifier. It relies on the mCGBlurb table (3.2.1) created before (that table contains multiple Swissprot/Unigene identifiers in a semicolon separated/terminated list. To find back which Swissprot/Unigene identifiers occur in each mCGBlurb field we search for each of them in turn. This is less than optimal and could be optimized. However, the query only takes 7 minutes, so I don't care that much (at the moment). The Swissprot2Ensmusg table was imported in section 5.3.1. The Unigene2Ensmusg table was imported in section 5.3.2. The query below is ran in the FkrpTaf4 database.

```
CREATE TABLE mCG2Ensmusg
SELECT DISTINCT
  Gene_ID,
  Ensembl
FROM Unigene2Ensmusg u2m, CG2Blurb blurb1
WHERE InStr(blurb1.UniGene,CONCAT(u2m.UniGene,";"))
UNION DISTINCT
SELECT DISTINCT Gene_ID, Ensembl
FROM Swissprot2Ensmusg s2m, CG2Blurb blurb2
WHERE InStr(blurb2.UniGene,CONCAT(s2m.swissprot,";"))
```

Example

```
+--------------+--------------------+
| Gene_ID      | Ensembl            |
+--------------+--------------------+
| mCG126572.1  | ENSMUSG00000062203 |
| mCG141162    | ENSMUSG00000049152 |
| mCG113184.1  | ENSMUSG00000050876 |
| mCG132220.1  | ENSMUSG00000030137 |
| mCG19273.2   | ENSMUSG00000028465 |
| mCG5925.1    | ENSMUSG00000053560 |
| mCG141342    | ENSMUSG00000051048 |
| mCG1031868.1 | ENSMUSG00000048355 |
| mCG1036470.1 | ENSMUSG00000038077 |
| mCG49016.1   | ENSMUSG00000025795 |
```

4 Linking the human genome to the mouse genome using Ensembl

In order to integrate the FKRP results we needed to map the mouse genes to the human genome, thereby respecting the function the different genes preform. inter-species genes with the same function are called orthologs. Ensembl provides a comparative database of genes between different species. However automatically mapping one onto the other was not as straightforward as one would expect. Theoretically one could write a query that would take all the stably annotated mouse genes, find them back in the homology table, determine the homology family and then find the human gene within that same family. The major problem that we encountered was that the Ensembl database has over 31'000'000 homology members, making straightforward joins of various tables a less than optimal solution. We optimized the querying using the following tricks.

1. clean out the 31'000'000 member set to the ones we actually need
2. work as quickly as possible with internal ids instead of the stable gene identifiers, such as ENSG....123123 and ENSMUSG...1234
3. Narrow the data-sets down as soon as possible using DISTINCT

4.1 What to map ?

Below we assume that we have a list of gene identifiers (form ENSMUSG...123) in the tomap.mouse_id column. The goal now is to create a new mapping from all these mouse_id's to human gene_ids. The tomap table is in our case defined as all potential mouse Ensembl id. Joining the stable_id's from Swissprot2mid (3.1.2) and Unigene2mid (3.1.3) provides us with the necessary things.

```
CREATE TABLE Tomap
  (mouse_id VARCHAR(32) PRIMARY KEY)
SELECT DISTINCT ensembl as mouse_id
FROM SwissProt2Mid
UNION DISTINCT
SELECT DISTINCT ensembl as mouse_id
FROM UniGene2Mid;
```

Example

```
+--------------------+
| mouse_id           |
+--------------------+
| ENSMUSG00000000028 |
| ENSMUSG00000000031 |
| ENSMUSG00000000037 |
| ENSMUSG00000000056 |
| ENSMUSG00000000058 |
| ENSMUSG00000000078 |
| ENSMUSG00000000088 |
| ENSMUSG00000000093 |
| ENSMUSG00000000103 |
```

4.2 Create a collection of mouse_homologs

This table will list all the homolog members that belong to the mouse family for which we are interested in the mapping. The tomap table is created in 4.1.

```
CREATE TABLE mouse_homolog
    (member_id int UNIQUE,
     stable_id VARCHAR(32) UNIQUE)
SELECT DISTINCT member_id, map.stable_id
FROM Tomap
JOIN mus_musculus_core_44_36e.gene_stable_id mus
ON mus.stable_id=Tomap.ensembl
JOIN ensembl_compara_44.member map
ON mus.stable_id=map.stable_id;
```

Example

```
+-----------+--------------------+
| member_id | stable_id          |
+-----------+--------------------+
|    828231 | ENSMUSG00000000028 |
|    338749 | ENSMUSG00000000031 |
|    682679 | ENSMUSG00000000037 |
|   1073599 | ENSMUSG00000000056 |
|    157512 | ENSMUSG00000000058 |
|    780907 | ENSMUSG00000000078 |
|    877703 | ENSMUSG00000000088 |
|   1052453 | ENSMUSG00000000093 |
|    667029 | ENSMUSG00000000103 |
|   1056719 | ENSMUSG00000000120 |
```

In the query above the ensembl_compara.member table maps a homology member to its stable gene identifier.

This table will be used as a starting point to find the homologies we are interested in. Afterward we will compare all the members of the homologies we like to a second table of human_homologs.

4.3 Create a collection of human_homologs

This table lists all possible target homologs (in our case, all stable gene members of the human genome)

```
CREATE TABLE human_homolog
    (member_id int UNIQUE,
     stable_id VARCHAR(32) UNIQUE)
SELECT DISTINCT member_id, map.stable_id
FROM homo_sapiens_core_44_36f.gene_stable_id hum
JOIN ensembl_compara_44.member map
ON hum.stable_id=map.stable_id;
```

Example

```
+-----------+-----------------+
| member_id | stable_id       |
+-----------+-----------------+
|         3 | ENSG00000198763 |
|         5 | ENSG00000198804 |
|         7 | ENSG00000198712 |
|         9 | ENSG00000198744 |
|        11 | ENSG00000198899 |
|        13 | ENSG00000198938 |
|        15 | ENSG00000198840 |
|        17 | ENSG00000198868 |
|        19 | ENSG00000198886 |
|        21 | ENSG00000198786 |
```

The mouse homologs and the human homologs are those that we are finally interested in and will be combined to filter out the (rather length) orthologs table.

4.4 Create a collection of homolog_members

```
CREATE TABLE homolog_member
    (member_id int PRIMARY KEY,
     stable_id VARCHAR(32) UNIQUE)
SELECT * FROM mouse_homolog
UNION DISTINCT
SELECT * FROM human_homolog;
```

Example

```
+-----------+--------------------+
| member_id | stable_id          |
+-----------+--------------------+
|    338098 | ENSG00000168394    |
|    338108 | ENSMUSG00000025147 |
|    338125 | ENSG00000204261    |
|    338142 | ENSG00000204259    |
|    338145 | ENSMUSG00000043186 |
|    338185 | ENSMUSG00000037887 |
|    338194 | ENSG00000204258    |
|    338215 | ENSG00000204257    |
|    338272 | ENSG00000204256    |
|    338283 | ENSMUSG00000073786 |
```

The mouse_homolog table was made in 4.2. The human homolog table was made in 4.3.

4.5 Select the homologs that could be interesting

The homologs that could be interesting are those that are no paralogs and those that have members in either the human gene or in the mouse gene. The first constraint is implemented by going through the homology compara table. The second constraint is implemented using the homolog_members table we made before.

```
CREATE TABLE ortologs
  (homology_id int,
   member_id int,
   INDEX (homology_id),
   INDEX (member_id))
SELECT h.homology_id, a.member_id
FROM homology h
JOIN ensembl_compara_44.homology_member b USING (homology_id)
JOIN homolog_member a USING (member_id)
WHERE description!="between_species_paralog"
AND description!="within_species_paralog";
```

Example

```
+-------------+-----------+
| homology_id | member_id |
+-------------+-----------+
|     3097693 |         1 |
|     3097793 |         1 |
|     3097840 |         1 |
|     3097935 |         1 |
|     3097966 |         1 |
|     3097987 |         1 |
|     3098140 |         1 |
|     3098283 |         1 |
|     3098381 |         1 |
|     3098403 |         1 |
|     3098478 |         1 |
|     3098500 |         1 |
|     5290964 |         3 |
|     5291000 |         3 |
|     5291061 |         3 |
|     5291114 |         3 |
|     5291145 |         3 |
|     5291260 |         3 |
|     5291274 |         3 |
|     5291346 |         3 |
|     5291503 |         3 |
|     5291563 |         3 |
|     5291571 |         3 |
|      712679 |         5 |
|      712702 |         5 |
|      712865 |         5 |
|      712903 |         5 |
|      712931 |          5 |
|      712987 |          5 |
|      713089 |          5 |
|      713115 |          5 |
```

The join itself requires the homology Ensembl table, the homolog_member table from Ensembl, and the homolog_member table we made ourselves (4.4). The last one will weed out all homologies that do not relate to mouse or human ids. The selection of only orthologs reduces the 31'000'000 row to around 5'000'000 and the further reduction using the human and mouse genome reduces it to around 3'000'000, which is a reasonable size to work with in the actual joining of the mouse to ortholog to human.

4.6 Find the orthologs that are really interesting

We are only interested in orthologs (4.5) that include a mouse homology (4.2).

```
CREATE TABLE Families
SELECT DISTINCT m.stable_id, b.homology_id
FROM mouse_homolog m
JOIN ortologs b USING (member_id);
```

Example

```
+--------------------+-------------+
| stable_id          | homology_id |
+--------------------+-------------+
| ENSMUSG00000000001 |    20625752 |
| ENSMUSG00000000001 |    20625974 |
| ENSMUSG00000000001 |    20636886 |
| ENSMUSG00000000001 |    20639601 |
| ENSMUSG00000000001 |    20640068 |
| ENSMUSG00000000001 |    20640705 |
| ENSMUSG00000000001 |    20643921 |
| ENSMUSG00000000001 |    20646072 |
| ENSMUSG00000000001 |    20646940 |
| ENSMUSG00000000001 |    20649601 |
```

4.7 Find the targets of the interesting ortologs

and check whether they occur in the human_homolog members. This is done by joining the previous calculated 'really interesting homologs' (4.6) with the orthologs (4.5) and the human_homologs (4.3).

```
SELECT DISTINCT
  b.stable_id as mouse,
  d.stable_id as human
FROM Families b
JOIN ortologs c ON b.homology_id=c.homology_id
JOIN human_homolog d ON c.member_id=d.member_id;
```

Example

```
+--------------------+-----------------+
| mouse              | human           |
+--------------------+-----------------+
| ENSMUSG00000000028 | ENSG00000093009 |
| ENSMUSG00000000056 | ENSG00000141562 |
| ENSMUSG00000000058 | ENSG00000105971 |
| ENSMUSG00000000078 | ENSG00000067082 |
| ENSMUSG00000000088 | ENSG00000178741 |
| ENSMUSG00000000093 | ENSG00000121068 |
| ENSMUSG00000000103 | ENSG00000005889 |
| ENSMUSG00000000103 | ENSG00000067646 |
| ENSMUSG00000000120 | ENSG00000064300 |
| ENSMUSG00000000125 | ENSG00000108379 |
| ENSMUSG00000000126 | ENSG00000143816 |
| ENSMUSG00000000127 | ENSG00000151422 |
| ENSMUSG00000000131 | ENSG00000169180 |
| ENSMUSG00000000142 | ENSG00000168646 |
| ENSMUSG00000000148 | ENSG00000106009 |
| ENSMUSG00000000149 | ENSG00000146535 |
| ENSMUSG00000000154 | ENSG00000110628 |
| ENSMUSG00000000157 | ENSG00000160255 |
| ENSMUSG00000000159 | ENSG00000183067 |
| ENSMUSG00000000168 | ENSG00000150768 |
| ENSMUSG00000000171 | ENSG00000204370 |
| ENSMUSG00000000182 | ENSG00000118972 |
| ENSMUSG00000000183 | ENSG00000111241 |
| ENSMUSG00000000184 | ENSG00000118971 |
| ENSMUSG00000000194 | ENSG00000148358 |
| ENSMUSG00000000204 | ENSG00000172123 |
| ENSMUSG00000000204 | ENSG00000205045 |
| ENSMUSG00000000216 | ENSG00000166828 |
| ENSMUSG00000000223 | ENSG00000102385 |
| ENSMUSG00000000244 | ENSG00000064201 |
```

The output of this query should be exported to ensmus2ensg.csv. This finalizes the mapping from the mouse genome to the human genome for the genes listed in the Tomap table.

5 Creating the Fkrp regulation tables

Our aim is to create a table that describes for each Ensembl gene the measured up/down regulations.

5.1 Create a ratio table for each gene

```
CREATE TABLE FkrpScrambledAverage
  (gene_id VARCHAR(64),
  signal FLOAT,
  INDEX (gene_id))
SELECT gene_id, avg(assay_normalized_signal) signal
FROM FkrpScrambled
GROUP BY gene_id;
```

Example

```
+------------------+----------+
| gene_id          | signal   |
+------------------+----------+
| 4930503K07       |  1.78667 |
| AB041802.1       |  1.55333 |
| AB045716.1       | 0.183333 |
| AB076245.1       | 0.283333 |
| AB080658.1       | 0.183333 |
| AB091827.1       |  8.20667 |
| AB099818.1_CDS_3 |  13.0667 |
| AF014450.1       |  1.05667 |
| AF045504.1       | 0.433333 |
| AF059259.1       |     0.23 |
```

The FkrpScrambled table was made in 2.2.3.

5.2 Calculating the Fkrp SiRNA averages

The above query make the scrambled average table, which contains for each gene (annotated as mCG... the average signal)

```
CREATE TABLE FkrpSiRna
  (gene_id VARCHAR(128),
  assay_normalized_signal FLOAT,
  INDEX (gene_id));
INSERT FkrpSiRna SELECT * FROM FkrpSiRna1;
INSERT FkrpSiRna SELECT * FROM FkrpSiRna2;
```

The FkrpSiRna1 and FkrpSiRna2 tables were made in 2.2.1 and 2.2.2 respectively. This query merges the two Fkrp SiRna tables. It was impossible to use one big sub-query (for some unknown reason), but this one works as well.

```
CREATE TABLE FkrpSiRnaAverage
  (gene_id VARCHAR(128),
  signal FLOAT,
  INDEX (gene_id))
SELECT gene_id, avg(assay_normalized_signal) signal
FROM FkrpSiRna GROUP BY gene_id;
```

Example

```
+------------------+--------+
| gene_id          | signal |
+------------------+--------+
| 4930503K07       |  1.746 |
| AB041802.1       |  1.844 |
| AB045716.1       |  0.202 |
| AB076245.1       |  0.376 |
| AB080658.1       |   0.27 |
| AB091827.1       |  6.316 |
| AB099818.1_CDS_3 | 18.152 |
| AF014450.1       |   0.93 |
| AF045504.1       |   0.44 |
| AF059259.1       |  0.442 |
```

5.2.1 Creating the mcg2Ratio table

This table calculates the average FkRp SiRNA signal intensity, based on the measurements of the scrambled SiRna (5.1) and non-scrambled SiRNA (5.2).

```
CREATE TABLE Fkrpmcg2Ratio
  (gene_id VARCHAR(64),
  ratio FLOAT,
  index (gene_id))
SELECT a.gene_id, b.signal/a.signal
FROM FkrpScrambledAverage a
JOIN FkrpSiRnaAverage b
USING (gene_id);
```

Example

```
+------------------+----------+
| gene_id          | ratio    |
+------------------+----------+
| 4930503K07       | 0.977239 |
| AB041802.1       |  1.18712 |
| AB045716.1       |  1.10182 |
| AB076245.1       |  1.32706 |
| AB080658.1       |  1.47273 |
| AB091827.1       | 0.769618 |
| AB099818.1_CDS_3 |  1.38918 |
| AF014450.1       | 0.880126 |
| AF045504.1       |  1.01538 |
| AF059259.1       |  1.92174 |
```

By joining the two tables we can calculate the final ratio. The next step now is to get rid of the mCG identifiers and replace them with Ensembl identifiers.

5.3 Importing the Swissprot|Unigen -> EnsG|EnsMusg tables

Before we can replace mCG identifiers with Ensembl identifiers we need to import the mappings we created early on. We first need the mouse to ensmusg table and then the ensmsug2enseg table. Both can be imported fairly straightforward from the imports as before:

5.3.1 Importing the Swissprot2Ensmusg table

This query imports the data from the csv file we obtained when querying the Ensembl database for the Swissprot2Ensmusg mapping (5.3.1).

```
CREATE TABLE Swissprot2Ensmusg
  (swissprot VARCHAR(128) KEY,
  ensembl VARCHAR(128),
  INDEX (ensembl));
LOAD DATA LOCAL INFILE 'imports/swissprot2ensmusg.csv'
INTO TABLE Swissprot2Ensmusg;
```

Example

```
+-----------+--------------------+
| swissprot | ensembl            |
+-----------+--------------------+
| P31254    | ENSMUSG00000069053 |
| P46425    | ENSMUSG00000038155 |
| Q8K2L9    | ENSMUSG00000033450 |
| P83882    | ENSMUSG00000049751 |
| P05531    | ENSMUSG00000054626 |
| Q9WV98    | ENSMUSG00000021079 |
| Q6IE32    | ENSMUSG00000060201 |
| O88574    | ENSMUSG00000031609 |
| Q9JI46    | ENSMUSG00000024213 |
| Q03740    | ENSMUSG00000070870 |
```

5.3.2 Importing the Unigene2Ensmusg table

This query imports the data from the csv file we obtained when querying the Ensembl database for the Unigen2Ensmusg mapping (3.1.3). The query below is ran in the FkrpTaf4 database.

```
CREATE TABLE Unigene2Ensmusg
  (unigene VARCHAR(128) KEY,
  ensembl VARCHAR(128),
  INDEX (ensembl));
LOAD DATA LOCAL INFILE 'imports/unigene2ensmusg.csv'
INTO TABLE Unigene2Ensmusg;
```

Example

```
+-----------+--------------------+
| unigene   | ensembl            |
+-----------+--------------------+
| Mm.10721  | ENSMUSG00000024963 |
| Mm.27038  | ENSMUSG00000036083 |
| Mm.39752  | ENSMUSG00000021573 |
| Mm.41636  | ENSMUSG00000002733 |
| Mm.76494  | ENSMUSG00000028528 |
| Mm.253378 | ENSMUSG00000043556 |
| Mm.260194 | ENSMUSG00000020474 |
| Mm.268582 | ENSMUSG00000071256 |
| Mm.317248 | ENSMUSG00000066443 |
| Mm.390885 | ENSMUSG00000014077 |
```

5.4 Importing the ensmusg2ensg table

The final join links the mcg2ensmusg table over the mouse to human orthologs. First, importing of the ensmusg2ensg table (downloaded from Ensembl using query 4.7).

```
CREATE TABLE Ensmusg2Ensg
  (mouse VARCHAR(128),
  human VARCHAR(128),
  INDEX (mouse),
  INDEX (human));
LOAD DATA LOCAL INFILE 'imports/ensmus2ensg.csv'
INTO TABLE Ensmusg2Ensg;
```

Example

```
+--------------------+-----------------+
| mouse              | human           |
+--------------------+-----------------+
| ENSMUSG00000000001 | ENSG00000065135 |
| ENSMUSG00000000028 | ENSG00000093009 |
| ENSMUSG00000000056 | ENSG00000141562 |
| ENSMUSG00000000058 | ENSG00000105971 |
| ENSMUSG00000000078 | ENSG00000067082 |
| ENSMUSG00000000088 | ENSG00000178741 |
| ENSMUSG00000000093 | ENSG00000121068 |
| ENSMUSG00000000103 | ENSG00000005889 |
| ENSMUSG00000000103 | ENSG00000067646 |
| ENSMUSG00000000120 | ENSG00000064300 |
```

5.5 Importing the Ensembl Gene descriptions

This statement imports the Ensembl genes descriptions (3.1.1) we obtained from the Ensembl database into the FkrpTaf4 database.

```
CREATE TABLE EnsgDescriptions
  (stable_id VARCHAR(128) PRIMARY KEY,
  description  VARCHAR(128));
LOAD DATA LOCAL INFILE 'imports/ensgdescriptions.csv'
INTO TABLE EnsgDescriptions;
```

5.6 The final join

The final join links the mCG identifiers in mcg2Ratio (5.2.1) to their Ensembl mouse identifier using the mcg2Ensmusg table (3.2.2). This joined table is then further linked to the Ensembl human genome identifiers through the ortholog mapping developed earlier (5.4) and then annotated with appropriate descriptions according to (5.5).

```
CREATE TABLE Ensg2Ratio
SELECT DISTINCT
  human,
  abs(log(ratio)/log(2)) as abslogratio,
  description
FROM mCG2Ratio
JOIN mCG2Ensmusg USING (Gene_id)
JOIN Ensmusg2Ensg ON (Ensembl=Mouse)
JOIN EnsgDescriptions ON (stable_id=human);
```

Example

```
+-----------------+-------------------+------------------------------------------------------------------
| human           | abslogratio       | description
```

```
+-----------------+--------------------+------------------------------------------------------------------
| ENSG00000168671 |  0.158794414944941 | UDP glycosyltransferase 3 family, polypeptide A2 [Source:RefSeq_pe
| ENSG00000145626 |  0.158794414944941 | UDP glycosyltransferase 3 family, polypeptide A1 [Source:RefSeq_pe
| ENSG00000183785 | 0.0131185176546133 | Peroxisome assembly protein 26 (Peroxin-26). [Source:Uniprot/SWISS
| ENSG00000137076 | 0.0896014658925754 | Talin-1. [Source:Uniprot/SWISSPROT;Acc:Q9Y490]
| ENSG00000160888 |  0.125966605983794 | Immediate early response gene 2 protein (Protein ETR101). [Source
| ENSG00000149380 |  0.378918118515268 | prolyl 4-hydroxylase, alpha III subunit precursor [Source:RefSeq_p
| ENSG00000151079 |  0.825970582585705 | Potassium voltage-gated channel subfamily A member 6 (Voltage-gate
| ENSG00000153179 |  0.211211462669622 | Ras association domain-containing protein 3. [Source:Uniprot/SWISS
| ENSG00000176040 |  0.050123519815702 | Transmembrane protease, serine 7 precursor (EC 3.4.21.-). [Source
| ENSG00000086102 |  0.183398227233556 | Transcriptional repressor NF-X1 (EC 6.3.2.-) (Nuclear transcriptio
```

6 Creating the TAF4 Regulation Tables

6.1 Averages

The average tables are based on the SiRNA and Scrambled measurements in the Hela (section 2.3.1 and 2.3.2) and Skndz cells (section 2.3.3 and 2.3.4).

6.1.1 Taf4ScrambledHelaAverage

```
 CREATE TABLE Taf4ScrambledHelaAverage
 (gene_id VARCHAR(128),
  signal FLOAT,
  INDEX (gene_id))
SELECT gene_id, avg(assay_normalized_signal) signal
FROM Taf4ScrambledHela
GROUP BY gene_id;
```

6.1.2 Taf4SiRnaHelaAverage

```
CREATE TABLE Taf4SiRnaHelaAverage
 (gene_id VARCHAR(128),
  signal FLOAT,
  INDEX (gene_id))
SELECT gene_id, avg(assay_normalized_signal) signal
FROM Taf4SiRnaHela
GROUP BY gene_id;
```

6.1.3 Taf4ScrambledSkndzAverage

```
CREATE TABLE Taf4ScrambledSkndzAverage
 (gene_id VARCHAR(128),
  signal FLOAT,
  INDEX (gene_id))
SELECT gene_id, avg(assay_normalized_signal) signal
FROM Taf4ScrambledSkndz
GROUP BY gene_id;
```

6.1.4 Taf4SiRnaSkndzAverage

```
CREATE TABLE Taf4SiRnaSkndzAverage
 (gene_id VARCHAR(128),
  signal FLOAT,
  INDEX (gene_id))
SELECT gene_id, avg(assay_normalized_signal) signal
FROM Taf4SiRnaSkndz
GROUP BY gene_id;
```

6.2 Ratios

6.2.1 Creating Taf4 hCG ratios for Hela cells

This ratio table is based on 6.1.1 and 6.1.2. The query is executed in the FkrpTaf4 schema

```
CREATE TABLE Taf4HcgRatioHela
  (gene_id VARCHAR(64),
  ratio FLOAT,
  index (gene_id))
SELECT a.gene_id, b.signal/a.signal
FROM Taf4ScrambledHela a
JOIN Taf4SiRnaHela b
USING (gene_id);
```

6.2.2 Creating Taf4 hCG ratios for SKNDZ cells

This ratio table is based on 6.1.3 and 6.1.4. This query is executed in the FkrpTaf4 schema.

```
CREATE TABLE Taf4EnsgRatioSkndz
  (gene_id VARCHAR(64),
  ratio FLOAT,
  index (gene_id))
SELECT a.gene_id, b.signal/a.signal
FROM Taf4ScrambledSkndz a
JOIN Taf4SiRnaSkndz b
USING (gene_id);
```

6.3 Abs-log ratios and Ensg identifiers

6.3.1 Taf4HcgAbsLogSkndz

```
CREATE TABLE Taf4HcgAbsLogSkndz
SELECT DISTINCT
  ensembl,
  log(abs(ratio)) as abslogratio,
  description
FROM Taf4HcgRatioSkndz
JOIN hCG2Ensg USING (Gene_ID)
JOIN EnsgDescriptions ON (stable_id=ensembl);
```

6.3.2 Taf4HcgAbsLogHela

```
CREATE TABLE Taf4EnsgAbsLogHela
SELECT DISTINCT
  ensembl,
  log(abs(ratio)) as abslogratio,
  description
FROM Taf4HcgRatioHela
JOIN hCG2Ensg USING (Gene_ID)
JOIN EnsgDescriptions ON (stable_id=ensembl);
```

6.3.3 Taf4Abslogratio

```
SELECT
  hela.ensembl,
  hela.abslogratio as hela,
  skndz.abslogratio as skndz,
  hela.description
FROM Taf4HcgAbsLogHela as hela
JOIN Taf4HcgAbsLogSkndz as skndz
USING (ensembl);
```

Example

```
+-----------------+---------------------+----------------------+------------------------------------------
| ensembl         | hela                | sknddz               | description
+-----------------+---------------------+----------------------+------------------------------------------
| ENSG00000115380 |  -0.079138526676969 |   -0.248584127949312 | EGF-containing fibulin-like extracellular
| ENSG00000177453 |   0.111473798056098 |   -0.307484718821447 | Serine/threonine-protein kinase NIM1 (EC
| ENSG00000163702 | -0.0600845596048174 | -0.00482188024916856 | Interleukin-17 receptor C precursor (IL-1
| ENSG00000164070 |  -0.189481181943389 |  -0.0204088316899511 | Heat shock 70 kDa protein 4L (Osmotic str
| ENSG00000164761 |  -0.122394168057243 |   -0.345501629149025 | Tumor necrosis factor receptor superfamil
| ENSG00000173918 |  -0.490711308082634 |   -0.171471503892196 | Complement C1q tumor necrosis factor-rela
| ENSG00000178562 |   0.186102216655366 |    -1.04400820819106 | T-cell-specific surface glycoprotein CD28
| ENSG00000009709 | -0.0957910545498839 |   0.0281143734785264 | Paired box protein Pax-7 (HUP1). [Source:
| ENSG00000132356 |   0.303738474652669 |   0.0885700945845682 | 5'-AMP-activated protein kinase catalytic
| ENSG00000125652 |   0.133938469086117 |  -0.0268717677139614 | Alkylated repair protein alkB homolog 7 p
```

7 Conclusion

We presented the steps necessary to

1. import all data from an Applied Biosystems 1700 scanner
2. use the Swissprot and Unigene gene identifiers to find back the Ensembl gene identifier
3. use Ensembl to perform an ortholog mapping from mouse genes to human genes

8 Online tables

Since the work we performed took a couple of weeks and we imagine that many people are interested in the actual mapping from mCG/hCG identifiers to their respective gene annotated identifiers we brought the mCG2ensmusg, hCG2Ensg and human2mouse mappings online at http://werner.onlinux.be/Papers/genemappings/index.