# metaGEM: an R package for meta-analysis of gene expression microarrays

Adaikalavan Ramasamy

April 23, 2009

## 1 The metaGEM package

The package "metaGEM" is a collection of R functions to meta-analyze multiple studies based on gene expression microarray (GEMs) technology. It currently has implementation to perform vote counting, Fisher's method of linear combination of *p*-values and random effects inverse-variance model.

This package was used in the following publication.

```
> citation("metaGEM")

To cite the metaGEM package in publications use:

  Ramasamy A, Mondry A, Holmes CC, Altman DG. Key Issues in Conducting a
  Meta-Analysis of Gene Expression Microarray Datasets. PLoS Med. 2008 Sep
  2;5(9):e184. PubMed ID: 18767902

A BibTeX entry for LaTeX users is

  @article{,
    title = {Key Issues in Conducting a Meta-Analysis of Gene Expression Microarray Datasets},
    author = {Adaikalavan Ramasamy and Adrian Mondry and Chris C. Holmes and Douglas G. Altman},
    journal = {PLoS Medicine},
    number = {9},
    volume = {5},
    pages = {e184},
    year = {2008},
  }

This free open-source software implements academic research by the authors and
co-workers. If you use it, please support the project by citing the
appropriate journal articles.
```

# 2 Installation

The package `metaGEM` depends on the following R packages (survival, splines, tools) which are part of the default R installation. It additionally depends on the `rmeta` package for the core meta-analysis functions which can be installed via:

```
   install.packages("rmeta")
```

It also requires the multtest package which is one of the default BioConductor packages. If not, you will need to install it using:

```
   source("http://bioconductor.org/biocLite.R")
   biocLite("multtest")
```

Next install the metaGEM package itself. In windows, one should modify the following command to reflect the path the package is in and excute it.

```
   install.packages("metaGEM_0.3.9.zip", repos=NULL)
```

Similarly, in a Linux or Unix environment, one should install the package using

```
   R CMD INSTALL metaGEM_0.3.9.tar.gz
```

# 3 Getting Started

We shall start by loading the package and the data to be used for illustrative purpose. The objective of the case study is to identify genes involved in the angiogenesis pathway that are consistently up- or down-regulated in cancers compared to normal tissue across different types of cancer.

```
> library(metaGEM)
> data(angio.GEMs)
> names(angio.GEMs)
```

```
 [1] "Bcells.Basso"         "Bcells.Haslinger"     "Bcells.Klein"
 [4] "Bladder.Dyrskjot"     "Bladder.Ramaswamy"    "BoneMarrow.Pellagati"
 [7] "Brain.Ramaswamy"      "Colon.Ramaswamy"      "Gastric.Chen"
[10] "Gastric.Hippo"        "HN.Ginos"             "HN.Winter"
[13] "HN.Kuriakose"         "Kidney.Lenburg"       "Kidney.Ramaswamy"
[16] "Liver.Chen"           "Lung.Beer"            "Lung.Bhatt"
[19] "Lung.Borczuk"         "Lung.Hu"              "Lung.Jones"
[22] "Pancreas.Coulevard"   "Pancreas.Gruetzmann"  "Pancreas.Ishikawa"
[25] "Pancreas.Ramaswamy"   "Prostate.Lapointe"    "Prostate.Singh"
[28] "Prostate.Welsh"       "Thyroid.Aldred"       "Thyroid.Huang"
```

`angio.GEMs` is a list containing the expression data for 30 gene expression cancer studies. Here, we combined the different studies in a list for progamming convenience but this needs not to be case. We identified probes involved in the angiogenesis pathway using gene ontology ID ”GO:0001525”.

Each study in turn is a list of three objects:

- *expr* : the gene expression with rows representing probes and columns representing individual samples

- *class*) : a vector used for two-class comparison and corresponds to the samples (same order as the columns in the *expr* matrix) with coding 0 (e.g. normal state) or 1 (e.g. diseased state).
- *keys* : a character vector that contains the mapping between gene-level ID to probe-level ID (same order as the rows in the *expr* matrix). If there are multiple gene-level IDs for the probe, separate by commas.

```
> study1 <- angio.GEMs[[1]]
> names(study1)
```

```
[1] "expr"  "class" "keys"
```

```
> dim(study1$expr)
```

```
[1] 81 76
```

```
> head(study1$expr[ , 1:17]) ## first six probes and seventeen arrays only
```

```
          GC_GSM44143 GC_GSM44144 GC_GSM44145 GC_GSM44146 GC_GSM44147 GC_GSM44148
1103_at      7.957125    7.989142    8.385903    7.866101    7.963237    7.875089
110_at      10.539679   10.736554   10.886344   10.642075   10.702474   10.687875
1114_at      8.923941    9.110819    9.334577    8.802456    8.853061    8.819157
1165_at      5.994917    5.820310    6.035808    6.080121    6.044072    6.067573
1262_s_at    6.100475    6.273634    6.209829    6.083056    6.193215    6.057253
1369_s_at    9.004378    9.348463    9.160286    8.959191    9.109953    9.981794
          GC_GSM44149 GC_GSM44150 GC_GSM44151 GC_GSM44152 BLL_GSM44194 BLL_GSM44195
1103_at      8.052673    7.794092    8.060426    7.897723     8.172545     8.015044
110_at      10.614298   10.467232   10.820543   10.635772    10.938416    10.565582
1114_at      8.997854    8.700318    9.104952    8.863807     8.924742     8.605557
1165_at      6.210225    6.071133    5.898282    6.018906     6.352702     7.379131
1262_s_at    6.304842    6.112066    6.256042    6.068427     6.552304     6.207223
1369_s_at   11.422599   12.586527   11.546107    9.901550    10.030576     8.869429
          BLL_GSM44196 BLL_GSM44197 BLL_GSM44198 BLL_GSM44199 DLBCL_GSM44204
1103_at       8.025910     8.135513     8.243668     8.022870       8.047781
110_at       10.425961    10.476280    10.438566    10.354283      10.545482
1114_at       8.501332     9.027549     8.415759     8.606003       8.848800
1165_at       6.431008     6.993048     7.797360     8.051576       6.837626
1262_s_at     6.543764     6.178579     6.235826     6.226617       6.477566
1369_s_at     8.833822     8.881111     9.230184     8.966454       9.065959
```

```
> study1$class
```

```
 [1] 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[43] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

```
> table(study1$class)
```

```
 0  1
10 66
```

```
> head( study1$keys )
```

```
[1] "Hs.283749,Hs.708689"        "Hs.513044"
[3] "Hs.68879"                   "Hs.83077"
[5] "Hs.133379"                  "Hs.624,Hs.551925,Hs.561078"
```

```
> class( study1$keys )
```

```
[1] "character"
```

For example, the first study is an investigation of the B-Cell carcinoma by Basso *et al* (PubMed ID:15778709) using 10 normal samples and 66 diseased cells (we grouped BLL and DLBCL together). It contains 81 probes that are involved in the angiogenesis pathway.

The probe-level needs to be mapped to gene-level ID such as UniGene Cluster ID (our chosen key) using tools such as the annotation packages within BioConductor and/or webtools such as SOURCE. Probes can map exactly to one, more than one or no gene-level IDs.

Before proceeding, we can summarize the relationship between probe-level and gene-level ID for each study. This can be used to help produce tables such as Table 4 in our paper. The table below shows that, for example, Basso *et al* contains 81 probes of which all were mapped to UniGene ID(s) (not surprising as we pre-selected only probes involved in angiogenesis). 73 out of these 81 were mapped to exactly one UniGene ID. Next, we "expand" the remaining nine probes and we end up with 92 records. Finally, "we summarize" to obtain 61 unique UniGene ID within this study. See Section 4.4 for more details on expanding and summarizing Gene IDs.

```
> t( sapply( angio.GEMs, function(study) summarize.probe2geneID( study$keys ) ) )
```

```
                     n.probes n.mapped n.one2one n.records n.geneIDs
Bcells.Basso               81       81        73        92        61
Bcells.Haslinger           81       81        73        92        61
Bcells.Klein               81       81        73        92        61
Bladder.Dyrskjot          140      140       114       178        79
Bladder.Ramaswamy          86       86        77        98        66
BoneMarrow.Pellagati      241      241       197       299        85
Brain.Ramaswamy            86       86        77        98        66
Colon.Ramaswamy            86       86        77        98        66
Gastric.Chen              118      118       103       133        80
Gastric.Hippo              51       51        44        60        51
```

```
HN.Ginos                  140    140    114    178     79
HN.Winter                 241    241    197    299     85
HN.Kuriakose               84     84     76     95     61
Kidney.Lenburg            190    190    152    240     83
Kidney.Ramaswamy           86     86     77     98     66
Liver.Chen                126    126    110    142     84
Lung.Beer                  51     51     44     60     51
Lung.Bhatt                 84     84     76     95     61
Lung.Borczuk               84     84     76     95     61
Lung.Hu                    62     62     62     62     36
Lung.Jones                101    101    101    101     63
Pancreas.Coulevard         62     62     62     62     36
Pancreas.Gruetzmann       190    190    152    240     83
Pancreas.Ishikawa         190    190    152    240     83
Pancreas.Ramaswamy         86     86     77     98     66
Prostate.Lapointe         133    133    117    149     85
Prostate.Singh             84     84     76     95     61
Prostate.Welsh             81     81     73     92     61
Thyroid.Aldred             84     84     76     95     61
Thyroid.Huang              81     81     73     92     61
```

The object angio.annDB is also loaded and contains some annotation information obtained from SOURCE. In a real application, this information would normally be obtained only after the meta-analysis.

# 4 Combining effect sizes

In general, meta-analysis is a two stage process. The first step is to summarize each study. In the case of meta-analysis, we need to summarize each probe within a study. These study-specific estimates are then combined in the second step.

## 4.1 Calculating study-specific effect size

We can use effect.sizes to calculate the effect size for each probe within a study. We choose the Hedges' adjusted g formulation for effect size since it corrects for the bias in effect size when sample sizes are small.

```
> all.ES <- lapply( angio.GEMs, effect.sizes )
> length(all.ES)
```

```
[1] 30
```

```
> head( all.ES[[1]] )
```

```
          n1        m1       sd1 n2        m2        sd2        diff pooled.sd
1103_at   66  8.255745 0.3034292 10  7.984151 0.16415891  0.27159371 0.2900847
110_at    66 10.646276 0.2004456 10 10.673285 0.12396966 -0.02700839 0.1927719
1114_at   66  8.778331 0.2558879 10  8.951094 0.18820387 -0.17276316 0.2486421
1165_at   66  7.590826 0.9194767 10  6.024135 0.10575551  1.56669181 0.8625393
1262_s_at 66  6.362857 0.3340360 10  6.165884 0.09253233  0.19697316 0.3147234
1369_s_at 66  9.217344 0.3759845 10 10.102085 1.29127487 -0.88474089 0.5718061
                   g      se.g                       keys
1103_at    0.9267354 0.3480096        Hs.283749,Hs.708689
110_at    -0.1386806 0.3395364                  Hs.513044
1114_at   -0.6877605 0.3441418                   Hs.68879
1165_at    1.7978999 0.3709183                   Hs.83077
1262_s_at  0.6194964 0.3432410                  Hs.133379
1369_s_at -1.5315394 0.3625285 Hs.624,Hs.551925,Hs.561078
```

`all.ES` is a list containing the following summary statistics for each probe within each study:

- *n1, m1, sd1* : The size, mean and standard deviation of group 1
- *n2, m2, sd2* : The size, mean and standard deviation of group 2
- *diff, pooled.sd* : The difference in group means (`diff = m2 - m1`) and the associated standard error.
- *g, se.g*: The effect size as formulated by the Hedges' adjusted *g* and the associated standard error.

## 4.2 Combining effect sizes across studies

Next, we calculate the random effects inverse-variance meta-analysis.

```
> output.REM <- combine.effect.sizes( all.ES )
> names(output.REM)
```

```
[1] "g"                "se.g"             "pooled.estimates"
```

```
> n.geneIDs <- apply(output.REM$g, 2, function(x) length(x[!is.na(x)]))
> names(n.geneIDs) <- NULL
> n.geneIDs    ## same as n.geneIDs column from summarize.probe2geneID above
```

```
 [1] 61 61 61 79 66 85 66 66 80 51 79 85 61 83 66 70 51 61 61 36 63 36 83 83 66 85 61 61
[29] 61 61
```

```
> head(output.REM$pooled.estimates)
```

```
          n.studies     summary se.summary       tau2     p.value
Hs.103527        20  0.24010957 0.15389738 0.34416960 0.118714346
Hs.111           27 -0.32605021 0.13525978 0.37135087 0.015928571
Hs.112408        29  0.13803905 0.10325414 0.17724565 0.181259752
Hs.11392         20 -0.71045064 0.22818338 0.89663822 0.001848779
Hs.115263        20  0.02441159 0.09926810 0.09180615 0.805747487
Hs.1239          27 -0.05388559 0.09800044 0.15097859 0.582421955
```

```
> table(output.REM$pooled.estimates$n.studies)
```

```
 1  2  3  4  5  6  7 10 11 12 13 14 15 16 17 20 23 24 27 28 29 30 
 7  7  9  1  2  1  3  2  3  3  2  2  2  4  1  4  3  7  7 11  8 20 
```

output.REM contains three components. The matrix g and se.g are the the study-specific effect size and the associated standard error for each gene-level ID. These are then used as inputs for random effects inverse-variance meta-analysis which is stored as the matrix component pooled.estimates.

## 4.3 Visualizing the results

We developed a simple plot plot.sumsum, motivated by the volcanno plot concept, to summarize all the results simultaneously. But first, we merge it with the annotation information to obtain the gene symbol (if available).

```
> pooled.ES   <- output.REM$pooled.estimates
> angio.annDB <- angio.annDB[ , "Symbol", drop=FALSE]
> pooled.ES <- merge(pooled.ES, angio.annDB, by=0, all.x=T)
> rownames(pooled.ES) <- pooled.ES$Row.names; pooled.ES$Row.names <- NULL
> mylabels <- paste( "\n", rownames(pooled.ES), "(", pooled.ES$Symbol, ")", sep="" )
> mylabels <- gsub("\\(NA\\)", "", mylabels)
```

Figure suggests that there appears to be five statistically significant up-regulated and three statistically significant down-regulated angiogenesis genes (at 1% false discovery rate) across all 30 studies.

However, if we inspect the results closer, we can see that the top few hits were identified only in few studies.

```
> pooled.ES$p.fdr <- p.adjust( pooled.ES$p.value, method="fdr" )
> pooled.ES <- pooled.ES[ order(pooled.ES$p.fdr), ]
> pooled.ES[ which(pooled.ES$p.fdr < 0.01), ]
```

```
          n.studies    summary se.summary       tau2      p.value Symbol        p.fdr
Hs.696701         1  1.1933454 0.23036503         NA 2.216043e-07   <NA> 2.415487e-05
Hs.696107         1  1.0204950 0.22669491         NA 6.743649e-06   <NA> 3.675289e-04
Hs.567968         3  0.7512201 0.19139650 0.07164694 8.675081e-05   <NA> 3.151946e-03
Hs.592212        27  0.3126040 0.08401733 0.12793127 1.986642e-04   TYMP 5.413601e-03
Hs.690402         1  0.8059848 0.22287456         NA 2.988257e-04   <NA> 6.514400e-03
Hs.152913        11 -0.5997692 0.16842910 0.24867897 3.695022e-04   EMCN 6.712623e-03
Hs.154210        28 -0.6547309 0.18731413 0.83611273 4.734174e-04  S1PR1 7.371785e-03
Hs.283749        30 -0.4416104 0.12857625 0.43231827 5.933880e-04    ANG 8.084912e-03
```

```
> plot.sumsum(pooled.ES, cex=0.5, labels=mylabels, lwd=2, signif.thr=0.05)
```

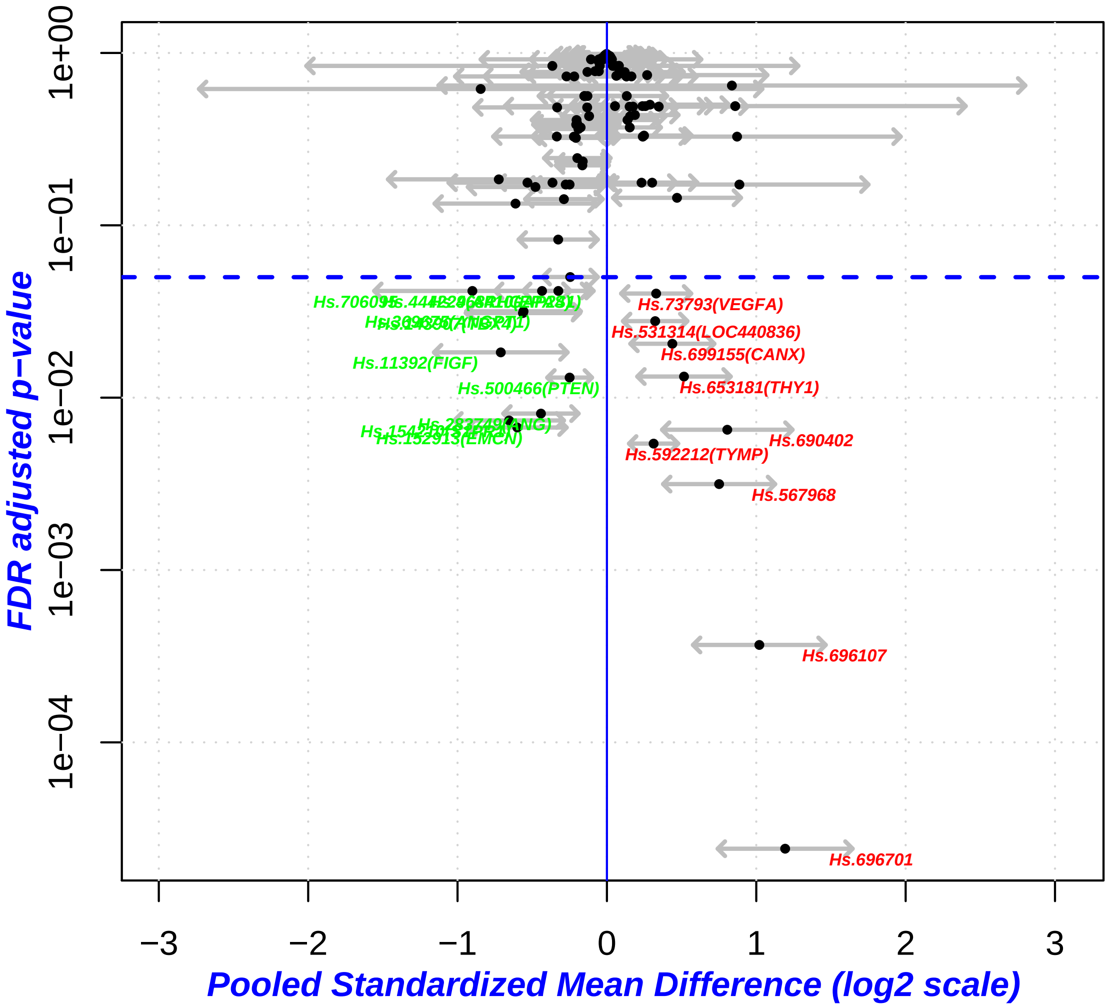


Figure 1: A Summary plot of the pooled effect size (black dots) with the 95% confidence interval (gray bars) sorted by the FDR. The Gene Symbol is given in parantheses when available.

Therefore we choose to impose an additional criteria that the probe must appear in at least five studies and relax the p-value threshold.

```
> length( w <- which(pooled.ES$p.fdr < 0.05 & pooled.ES$n.studies >=5) )
```

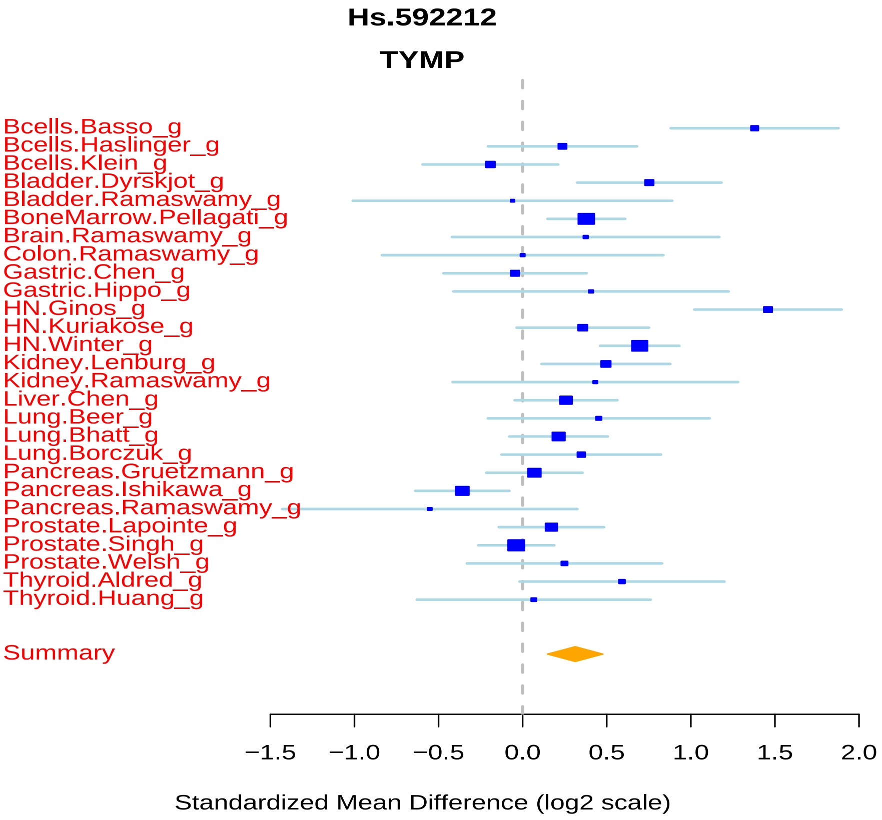
```
[1] 14
```

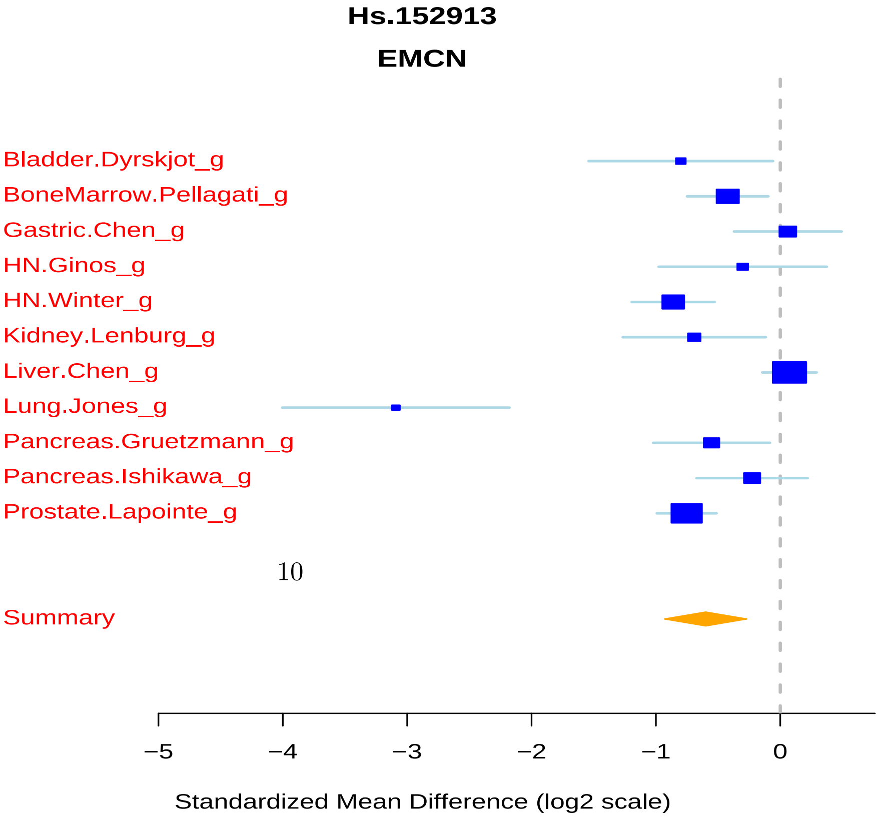
```
> pooled.ES[w, ]
```

```
          n.studies    summary se.summary      tau2      p.value    Symbol       p.fdr
Hs.592212        27  0.3126040 0.08401733 0.1279313 0.0001986642      TYMP 0.005413601
Hs.152913        11 -0.5997692 0.16842910 0.2486790 0.0003695022      EMCN 0.006712623
Hs.154210        28 -0.6547309 0.18731413 0.8361127 0.0004734174     S1PR1 0.007371785
Hs.283749        30 -0.4416104 0.12857625 0.4323183 0.0005933880       ANG 0.008084912
Hs.500466        30 -0.2495422 0.07634018 0.1240698 0.0010799759      PTEN 0.013079708
Hs.653181        30  0.5162059 0.15957086 0.6474617 0.0012165844      THY1 0.013260770
Hs.11392         20 -0.7104506 0.22818338 0.8966382 0.0018487793      FIGF 0.018319722
Hs.699155        30  0.4381103 0.14348038 0.5058356 0.0022622580      CANX 0.020548843
Hs.531314        24  0.3231964 0.11005045 0.1858239 0.0033161615 LOC440836 0.027804739
Hs.143907        10 -0.5622203 0.19522056 0.2585621 0.0039777178      TBX4 0.030969375
Hs.369675        29 -0.5583001 0.19584308 1.0148108 0.0043615979    ANGPT1 0.031694278
Hs.73793         30  0.3296404 0.11972405 0.3630968 0.0058991564     VEGFA 0.040188003
Hs.444229        13 -0.4342368 0.16080713 0.2764305 0.0069264923  ARHGAP24 0.041612265
Hs.468410        30 -0.3252423 0.12058280 0.3680008 0.0069914199     EPAS1 0.041612265
```

Next, we can use forest plots to identify the contribution of each study, identify possible outliers and subgroup effect. Plotting the top three hits, for example, shows that the summary effect size for Hs.154210 (S1PR1) appears to be largely due to the lung cancer studies. Proving that such a subgrouping effect exists would require further work.

```
> tophits <- rownames(pooled.ES[w, ])
> tophits.symbol <- as.character(pooled.ES[w, "Symbol"])
> for(i in 1:3){
+     forest.plot( output.REM, key=tophits[i] )
+     title( main=tophits.symbol[i], line=0 )
+ }
```


Hs.592212
TYMP
Bcells.Basso_g
Bcells.Haslinger_g
Bcells.Klein_g
Bladder.Dyrskjot_g
Bladder.Ramaswamy_g
BoneMarrow.Pellagati_g
Brain.Ramaswamy_g
Colon.Ramaswamy_g
Gastric.Chen_g
Gastric.Hippo_g
HN.Ginos_g
HN.Kuriakose_g
HN.Winter_g
Kidney.Lenburg_g
Kidney.Ramaswamy_g
Liver.Chen_g
Lung.Beer_g
Lung.Bhatt_g
Lung.Borczuk_g
Pancreas.Gruetzmann_g
Pancreas.Ishikawa_g
Pancreas.Ramaswamy_g
Prostate.Lapointe_g
Prostate.Singh_g
Prostate.Welsh_g
Thyroid.Aldred_g
Thyroid.Huang_g
Summary
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
Standardized Mean Difference (log2 scale)


Hs.152913
EMCN
Bladder.Dyrskjot_g
BoneMarrow.Pellagati_g
Gastric.Chen_g
HN.Ginos_g
HN.Winter_g
Kidney.Lenburg_g
Liver.Chen_g
Lung.Jones_g
Pancreas.Gruetzmann_g
Pancreas.Ishikawa_g
Prostate.Lapointe_g
Summary
−5
−4
−3
−2
−1
0
Standardized Mean Difference (log2 scale)

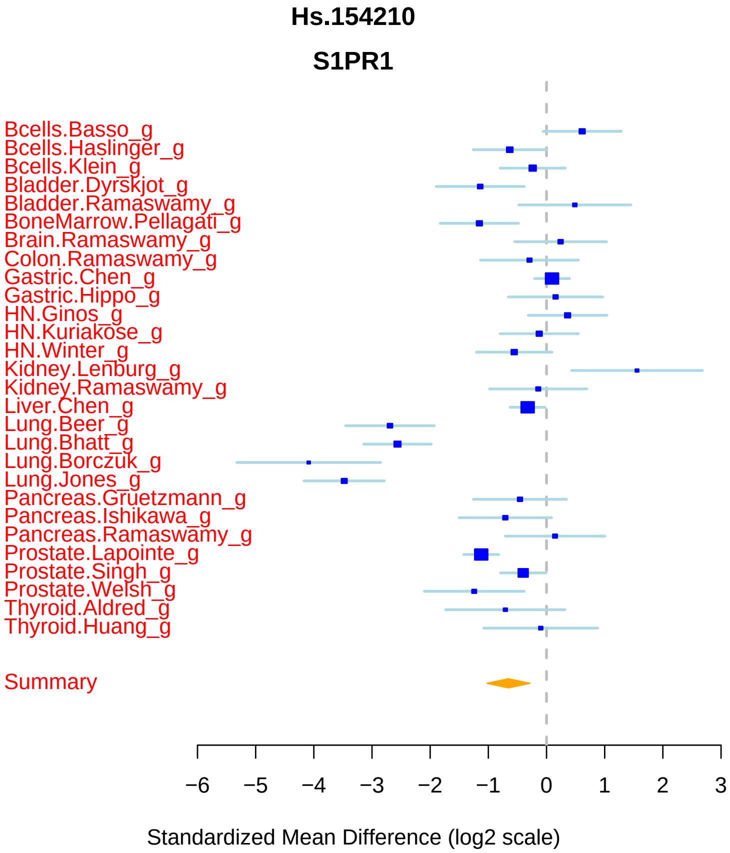


Figure 2: Forest plots for the three most statistically significant genes identified from the meta-analysis that was present in at least 5/30 studies.

### 4.4 ASIDE: How the function `summarize.effect.sizes` works

This section may be safely skipped and details what happens to each study in the first few steps of the `combine.effect.sizes` function. First, we create and artificial example:

```
> expr1 <- matrix(round(rnorm(35), digits=2), nr=5, nc=7 )
> rownames(expr1) <- paste("probe", 1:5, sep="")
> colnames(expr1) <- c(paste("normal", 1:3, sep=""), paste("tumor", 1:4, sep=""))
> study1 <- list(expr  = expr1,
+                class = rep( 0:1, c(3,4) ),
+                keys  = c("A", "B", "A,C", NA, "E") )
> study1
```

```
$expr
       normal1 normal2 normal3 tumor1 tumor2 tumor3 tumor4
probe1   -1.36   -0.34    0.44   0.16   0.74  -0.06  -0.36
probe2    0.49    0.83    0.55  -0.79  -0.47  -0.39  -1.79
probe3    0.15   -0.30    0.10   0.25  -0.88  -1.86  -0.71
probe4    0.34   -0.72   -0.97  -1.80  -0.87  -1.00  -0.80
probe5   -0.43   -0.55    0.85   1.97   1.28  -2.68  -2.43

$class
[1] 0 0 0 1 1 1 1

$keys
[1] "A"   "B"   "A,C" NA    "E"
```

```
> es1 <- effect.sizes(study1)
> es1[ , c("g", "se.g", "keys")]
```

```
                g      se.g keys
probe1  0.6736436 0.8108532    A
probe2 -2.4416062 1.2479683    B
probe3 -0.9596957 0.8566365  A,C
probe4 -0.9908651 0.8624154 <NA>
probe5 -0.1824157 0.7673138    E
```

First, discard any probes that could not be mapped to Gene IDs.

```
> bad <- which(is.na(es1$g) | is.na(es1$keys) | es1$keys == "NA")
> es1.clean <- es1[ setdiff(1:nrow(es1), bad), ]
> es1.clean[ , c("g", "se.g", "keys")]
```

```
                g      se.g keys
probe1  0.6736436 0.8108532    A
probe2 -2.4416062 1.2479683    B
probe3 -0.9596957 0.8566365  A,C
probe5 -0.1824157 0.7673138    E
```

Next, we need to replace probes that map to multiple Gene IDs with a new record for each Gene ID. This happens for probe 3 in this example.

```
> es1.expanded <- expand.df(es1.clean)
> es1.expanded[ , c("g", "se.g", "keys")]
```

```
                  g      se.g keys
probe1    0.6736436 0.8108532    A
probe2   -2.4416062 1.2479683    B
probe3   -0.9596957 0.8566365    A
probe3.1 -0.9596957 0.8566365    C
probe5   -0.1824157 0.7673138    E
```

Finally, if a Gene ID is identified multiple times within a study, we need to summarize it to get a single estimate per Gene ID. Here, we use the fixed effect inverse variance (the default) to summarize.

```
> es1.summarized <- summ.eff.within(es1.expanded)
> es1.summarized
```

```
           g      se.g
A -0.0982142 0.5888803
B -2.4416062 1.2479683
C -0.9596957 0.8566365
E -0.1824157 0.7673138
```

Here Gene A is identified twice and the output for Gene A should be identical to:

```
> w <- which(es1.expanded$keys == "A")
> meta.summaries( es1.expanded$g[w], es1.expanded$se.g[w] )
```

```
Fixed-effects meta-analysis
Call: meta.summaries(d = es1.expanded$g[w], se = es1.expanded$se.g[w])
Summary effect=-0.0982   95% CI (-1.25, 1.06)
Estimated heterogeneity variance: 0.64  p= 0.166
```

We can repeat the steps above for the other studies and combine the effect sizes using the `multimerge` function which is equivalent to an iterative `merge` on the rownames (i.e. using `by=0` argument).

# 5 Vote counting and combining $p$-values

Vote counting involves counting the number of studies in which a gene was declared as significant. We can also combine the $p$-values using Fisher's sum of logs methods for a given gene which accounts for the strength of signficance. For both methods, we need to first calculate a measure of significance for each gene in each study.

## 5.1 Calculating study-specific significance

We can use `ttest.Pvalues` or `ttest.Qvalues` to calculate the nominal $p$-values or the $Q$-values (the FDR-adjusted $p$-values) from a $t$-test statistics with equal variance assumption. These functions return the one-sided significance testing for up- and down-regulation of the gene. Here we take the $Q$-values as the significance measure.

```
> all.Qvals <- lapply( angio.GEMs, ttest.Qvalues )
> length(all.Qvals)
```

```
[1] 30
```

```
> head( all.Qvals[[1]] )
```

```
                  Q.up       Q.down                        keys
1103_at   1.696097e-02 0.9999995329        Hs.283749,Hs.708689
110_at    1.000000e+00 0.7256864230                  Hs.513044
1114_at   1.000000e+00 0.0941102585                   Hs.68879
1165_at   3.002087e-05 0.9999995329                   Hs.83077
1262_s_at 9.999838e-02 0.9999995329                  Hs.133379
1369_s_at 1.000000e+00 0.0001609331 Hs.624,Hs.551925,Hs.561078
```

## 5.2 Fisher's sum of logs method

This method sums the logarithm of the one-sided hypothesis testing significance ($p$-values or $Q$-values) across $k$ studies for a given gene. One advantage is that we can account for the strength in evidence rather than choosing an arbitary threshold for declaring significance as we do in vote counting.

The p-value for the test statistic can be calculated using permutation method or by using the asymptotic distribution of chi-square with $2k$ degrees of freedom. Our function `sum.of.logs` calculates the Fisher statistics (minus twice the sum of log of significances) and calculates the asymptotical $p$-values for both alternative hypotheses.

```
> output.Fisher <- sum.of.logs(all.Qvals)
> head(output.Fisher)
```

```
          F.stat.up   F.pval.up F.stat.down  F.pval.down
Hs.103527 73.852309 0.000891894    25.03625 9.690188e-01
Hs.111    25.743055 0.999610064   150.65753 4.570851e-11
Hs.112408 47.127637 0.845547939    32.28585 9.975224e-01
Hs.11392   7.627488 0.999999995   243.75721 4.888361e-31
Hs.115263 23.673572 0.981231036    20.80779 9.947378e-01
Hs.1239   43.803800 0.837798858    56.96026 3.655263e-01
```

### 5.3 Vote counting

In this method, we count the number of studies in which a gene reaches a certain statistical significance. However, to do this, we need to choose a value for the statistical significance. We choose a threshold of 0.10 for this exercise. The function `count.votes` implements vote counting procedure and returns two ordered list - one each for testing the alternative hypothesis that genes are up-regulated and down-regulation.

```
> output.VC <- count.votes( all.Qvals, vote.threshold=0.10 )
> head(output.VC)
```

```
          yes.down valid.down yes.up valid.up
Hs.103527        1         20      7       20
Hs.111           9         27      2       27
Hs.112408        2         29      5       29
Hs.11392         7         20      0       20
Hs.115263        1         20      2       20
Hs.1239          4         27      3       27
```

The output above shows that, for example, the gene Hs.103527 was significantly up-regulated in 1/20 studies and was significantly down-regulated in 7/20 studies.

## 6 Combining ranks

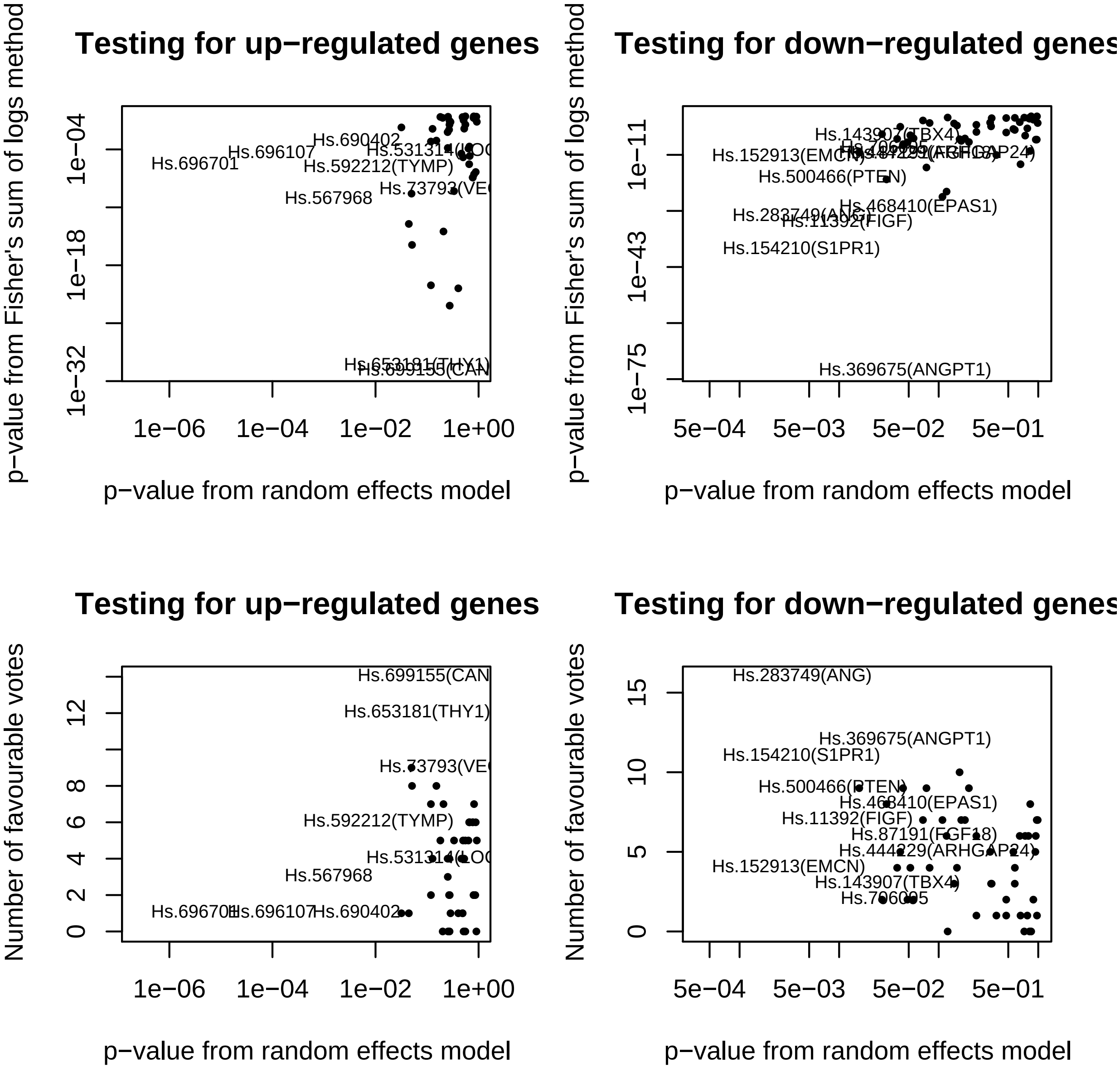
Not yet implemented in this package. However, users may be interested in the RankProd package.

## 7 Comparison of results

To conclude, we can do a quick and dirty comparison of the outputs. For each alternative hypothesis, we plot the *p*-value from the random effects model against the Fisher's sum of logs method and also vote counting as follows. We label the points which are significant at the nominal p-value of 0.01 using the random effects model.

```
> out <- cbind( pooled.ES,
+              output.VC[ rownames(pooled.ES), ],
+              output.Fisher[ rownames(pooled.ES), ] )
> rownames(out) <- paste( rownames(out), "(", out$Symbol, ")", sep="" )
> rownames(out) <- gsub("\\(NA\\)", "", rownames(out))
> out.pos <- out[ which(out$summary > 0), ]
> out.neg <- out[ which(out$summary < 0), ]
> myplot <- function(x, y, x.thres=0.01, labels, cex=0.75, ...){
+     plot(x, y, type="n", ...)
+     w <- which( x < x.thres )
+     points(x[-w], y[-w], cex=cex, pch=20)
+     text(x[w], y[w], labels=labels[w], cex=cex, pos=4)
+ }
> lab1 <- "p-value from random effects model"
> lab2 <- "p-value from Fisher's sum of logs method"
> lab3 <- "Number of favourable votes"
> main1 <- "Testing for up-regulated genes"
> main2 <- "Testing for down-regulated genes"

> par(mfrow=c(2,2))
> myplot( out.pos$p.value, out.pos$F.pval.up, labels=rownames(out.pos),
+         log="xy", xlab=lab1, ylab=lab2, main=main1 )
> myplot( out.neg$p.value, out.neg$F.pval.down, labels=rownames(out.neg),
+         log="xy", xlab=lab1, ylab=lab2, main=main2 )
> myplot( out.pos$p.value, out.pos$yes.up, labels=rownames(out.pos),
+         log="x", xlab=lab1, ylab=lab3, main=main1 )
> myplot( out.neg$p.value, out.neg$yes.down, labels=rownames(out.neg),
+         log="x", xlab=lab1, ylab=lab3, main=main2 )
```



As you can see for the alternative hypothesis for up-regulated genes (left column of figures), the genes CANX and THY1 have a significant $p$-values for random effects model and Fisher's method as well as having a high number of favourable votes; making them worthy of further investigation. Similarly, we can prioritize ANG, ANGPT1 and S1PR1 for investigation of the down-regulated alternative hypothesis.

```
> w <- which( out$Symbol %in% c("CANX", "THY1", "ANG", "ANGPT1", "S1PR1") )
> out[ w, ]
                  n.studies    summary se.summary      tau2      p.value Symbol
Hs.154210(S1PR1)         28 -0.6547309  0.1873141 0.8361127 0.0004734174  S1PR1
Hs.283749(ANG)           30 -0.4416104  0.1285763 0.4323183 0.0005933880    ANG
Hs.653181(THY1)          30  0.5162059  0.1595709 0.6474617 0.0012165844   THY1
Hs.699155(CANX)          30  0.4381103  0.1434804 0.5058356 0.0022622580   CANX
Hs.369675(ANGPT1)        29 -0.5583001  0.1958431 1.0148108 0.0043615979 ANGPT1
                        p.fdr yes.down valid.down yes.up valid.up F.stat.up
Hs.154210(S1PR1)  0.007371785       11         28      1       28  19.81023
Hs.283749(ANG)    0.008084912       16         30      2       30  39.66767
Hs.653181(THY1)   0.013260770        3         30     12       30 284.55167
Hs.699155(CANX)   0.020548843        4         30     14       30 287.93036
Hs.369675(ANGPT1) 0.031694278       12         29      4       29  87.86476
                     F.pval.up F.stat.down  F.pval.down
Hs.154210(S1PR1)  9.999981e-01   320.97880 7.758202e-39
Hs.283749(ANG)    9.801849e-01   275.44456 2.367018e-29
Hs.653181(THY1)   6.348750e-31    44.96753 9.257646e-01
Hs.699155(CANX)   1.645901e-31    54.10322 6.899125e-01
Hs.369675(ANGPT1) 6.896912e-03   509.49172 2.006653e-73
```