categoryCompare: High-throughput data meta-analysis using feature annotations

R.M. Flight
rflight79@gmail.com

October 14, 2013

1 Introduction

Current high-throughput molecular biology experiments are generating larger and larger amounts of data. Although there are many different methods to analyze individual experiments, methods that allow the comparison of different data sets are sorely lacking. This is important due to the number of experiments that have been carried out on biological systems that may be amenable to either fusion or comparison. Most of the current tools available focus on finding those genes in experiments that are listed as the same, or that can be shown statistically that it is significant that the gene was listed in the results of both experiments.

However, what many of these tools do not do is consider the similarities (and just as importantly, the differences) between experimental results at the categorical level. Categoical data includes any gene annotation, such as Gene Ontologies, KEGG pathways, chromosome location, etc. categoryCompare has been developed to allow the comparison of high-throughput experiments at a categorical level, and to explore those results in an intuitive fashion.

2 Sample Data

To make the concept more concrete, we will examine data from the microarray data set estrogen available from Bioconductor. This data set contains 8 samples, with 2 levels of estrogen therapy (present vs absent), and two time points (10 and 48 hours). A pre-processed version of the data is available with this package, the commands used to generate it are below. Note: the preprocessed one keeps only the top 100 genes, if you use it the results will be slightly different than those shown in the vignette.

```
> library("affy")
> library("hgu95av2.db")
> library("genefilter")
> library("estrogen")
> library("limma")

> datadir <- system.file("extdata", package = "estrogen")
> pd <- read.AnnotatedDataFrame(file.path(datadir,"estrogen.txt"),
```

```
                header = TRUE, sep = "", row.names = 1)
> pData(pd)

             estrogen time.h
low10-1.cel    absent     10
low10-2.cel    absent     10
high10-1.cel  present     10
high10-2.cel  present     10
low48-1.cel    absent     48
low48-2.cel    absent     48
high48-1.cel  present     48
high48-2.cel  present     48
```

Here you can see the descriptions for each of the arrays. First, we will read in the cel files, and then normalize the data using RMA.

```
> currDir <- getwd()
> setwd(datadir)
> a <- ReadAffy(filenames=rownames(pData(pd)), phenoData = pd, verbose = TRUE)

1 reading low10-1.cel ...instantiating an AffyBatch (intensity a 409600x8 matrix)...done.
Reading in : low10-1.cel
Reading in : low10-2.cel
Reading in : high10-1.cel
Reading in : high10-2.cel
Reading in : low48-1.cel
Reading in : low48-2.cel
Reading in : high48-1.cel
Reading in : high48-2.cel

> eData <- rma(a)

Background correcting
Normalizing
Calculating Expression

> setwd(currDir)
```

To make it easier to conceptualize, we will split the data up into two eSet objects by time, and perform all of the manipulations for calculating significantly differentially expressed genes on each eSet object.

So for the 10 hour samples:

```
> e10 <- eData[, eData$time.h == 10]
> e10 <- nsFilter(e10, remove.dupEntrez=TRUE, var.filter=FALSE,
               feature.exclude="^AFFX")$eset
> e10$estrogen <- factor(e10$estrogen)
```

```
> d10 <- model.matrix(~0 + e10$estrogen)
> colnames(d10) <- unique(e10$estrogen)
> fit10 <- lmFit(e10, d10)
> c10 <- makeContrasts(present - absent, levels=d10)
> fit10_2 <- contrasts.fit(fit10, c10)
> eB10 <- eBayes(fit10_2)
> table10 <- topTable(eB10, number=nrow(e10), p.value=1, adjust.method="BH")
> table10$Entrez <- unlist(mget(rownames(table10), hgu95av2ENTREZID, ifnotfound=NA))
```

And the 48 hour samples we do the same thing:

```
> e48 <- eData[, eData$time.h == 48]
> e48 <- nsFilter(e48, remove.dupEntrez=TRUE, var.filter=FALSE,
              feature.exclude="^AFFX" )$eset
> e48$estrogen <- factor(e48$estrogen)
> d48 <- model.matrix(~0 + e48$estrogen)
> colnames(d48) <- unique(e48$estrogen)
> fit48 <- lmFit(e48, d48)
> c48 <- makeContrasts(present - absent, levels=d48)
> fit48_2 <- contrasts.fit(fit48, c48)
> eB48 <- eBayes(fit48_2)
> table48 <- topTable(eB48, number=nrow(e48), p.value=1, adjust.method="BH")
> table48$Entrez <- unlist(mget(rownames(table48), hgu95av2ENTREZID, ifnotfound=NA))
```

And grab all the genes on the array to have a background set.

```
> gUniverse <- unique(union(table10$Entrez, table48$Entrez))
```

For both time points we have generated a list of genes that are differentially expressed in the present vs absent samples. To compare the time-points, we could find the common and discordant genes from both experiments, and then try to interpret those lists. This is commonly done in many meta-analysis studies that attempt to combine the results of many different experiments.

An alternative approach, used in categoryCompare, would be to compare the significantly enriched categories from the two gene lists. Currently the package supports two category classes, Gene Ontology, and KEGG pathways. Both are used below.

Note 1: I am not proposing that this is the best way to analyse **this** particular data, as it is a sample data set that merely serves to illustrate the functionality of this package. However, there are many different experiments where this type of approach is definitely appropriate, and it is up to the user to determine if their data fits the analytical paradigm advocated here.

Note 2: If you are viewing this document as a vignette, some of the commands below are not actually evaluated (specifically those involving passing data to Cytoscape), however the commands themselves will work if used in an R session.

3 Create Gene List

```
> library("categoryCompare")
> library("GO.db")
```

```
> library("KEGG.db")
> # load the saved data with the package
> # data(ccData)
>
> g10 <- unique(table10$Entrez[table10$adj.P.Val < 0.05])
> g48 <- unique(table48$Entrez[table48$adj.P.Val < 0.05])
```

For each list the genes of interest, and a background must be defined. Here we are using those genes with an adjusted P-value of less than 0.05 as the genes of interest, and all of the genes on the chip as the background.

```
> list10 <- list(genes=g10, universe=gUniverse, annotation='org.Hs.eg.db')
> list48 <- list(genes=g48, universe=gUniverse, annotation='org.Hs.eg.db')
> geneLists <- list(T10=list10, T48=list48)
> geneLists <- new("ccGeneList", geneLists, ccType=c("BP","KEGG"))
> fdr(geneLists) <- 0 # this speeds up the calculations for demonstration
> geneLists

List:  T10
Size of gene list:  671
Size of gene universe:  8656
Annotation:  org.Hs.eg.db

List:  T48
Size of gene list:  110
Size of gene universe:  8656
Annotation:  org.Hs.eg.db

Types of annotations to examine:  BP; KEGG
Number of FDR runs to perform:  0
pValue Cutoff to decide significantly enriched annotations:  0.05
Testdirection:  over represented
```

4 Annotation Enrichment

Now run the enrichment calculations on each list. In this case enrichment will be performed using the biological process (BP) Gene Ontology, and KEGG Pathways.

```
> enrichLists <- ccEnrich(geneLists)

Performing Enrichment Calculations ....
T10 : BP
T48 : BP
T10 : KEGG
T48 : KEGG
Done!!
```

```
> enrichLists
```

```
    Annotation category:  GO   BP
                FDR runs:  0
Default p-values to use:  pval
                 pCutoff:  0.05

List:  T10
Gene to GO BP  test for over-representation
5364 GO BP ids tested (596 have p <= 0.05 & count >= 0)
Selected gene set size: 633
    Gene universe size: 7868
    Annotation package: org.Hs.eg

List:  T48
Gene to GO BP  test for over-representation
2302 GO BP ids tested (518 have p <= 0.05 & count >= 0)
Selected gene set size: 106
    Gene universe size: 7868
    Annotation package: org.Hs.eg


    Annotation category:  KEGG
                FDR runs:  0
Default p-values to use:  pval
                 pCutoff:  0.05

List:  T10
Gene to KEGG  test for over-representation
192 KEGG ids tested (24 have p <= 0.05 & count >= 0)
Selected gene set size: 331
    Gene universe size: 3655
    Annotation package: org.Hs.eg

List:  T48
Gene to KEGG  test for over-representation
73 KEGG ids tested (9 have p <= 0.05 & count >= 0)
Selected gene set size: 64
    Gene universe size: 2758
    Annotation package: org.Hs.eg
```

There are a lot of GO BP processes enriched using the p-value cutoff of 0.05, so lets make that more stringent. This is done here merely for speed, in a usual analysis you should choose this number, and the type of cutoff (p-value or fdr) carefully.

```
> pvalueCutoff(enrichLists$BP) <- 0.001
> enrichLists
```

```
    Annotation category:  GO   BP
               FDR runs:  0
Default p-values to use:  pval
                pCutoff:  0.001

List:  T10
Gene to GO BP  test for over-representation
5364 GO BP ids tested (152 have p <= 0.001 & count >= 0)
Selected gene set size: 633
    Gene universe size: 7868
    Annotation package: org.Hs.eg

List:  T48
Gene to GO BP  test for over-representation
2302 GO BP ids tested (143 have p <= 0.001 & count >= 0)
Selected gene set size: 106
    Gene universe size: 7868
    Annotation package: org.Hs.eg


    Annotation category:  KEGG
               FDR runs:  0
Default p-values to use:  pval
                pCutoff:  0.05

List:  T10
Gene to KEGG  test for over-representation
192 KEGG ids tested (24 have p <= 0.05 & count >= 0)
Selected gene set size: 331
    Gene universe size: 3655
    Annotation package: org.Hs.eg

List:  T48
Gene to KEGG  test for over-representation
73 KEGG ids tested (9 have p <= 0.05 & count >= 0)
Selected gene set size: 64
    Gene universe size: 2758
    Annotation package: org.Hs.eg
```

Currently you can see that for T10, there are 141 processes enriched, and 123 for T48. For KEGG, there are much smaller numbers of pathways enriched.

To see which processes and pathways are enriched, and to compare them, we will run `ccCompare`, after generating a **ccOptions** object to tell the software exactly which comparisons to do.

```
> ccOpts <- new("ccOptions", listNames=names(geneLists), outType='none')
> ccOpts

List Names:  T10; T48
Comparisons:  T10; T48; T10,T48
Colors:  #FF7A9E; #89BC00; #00C8EA
Output Types:  none

> ccResults <- ccCompare(enrichLists,ccOpts)
> ccResults

ccCompare results for:

Annotation category:  GO   BP
Main graph: A graphNEL graph with directed edges
Number of Nodes = 142
Number of Edges = 6942

Annotation category:  KEGG
Main graph: A graphNEL graph with directed edges
Number of Nodes = 24
Number of Edges = 82
```

The `ccResults` is a list object where for each type of category (Gene Ontologies, KEGG pathways, etc) there are **ccCompareResult** objects containing various pieces, including the output of the enrichments in table form (mainTable) with a designation as to which of the geneLists they originated from, a graph that shows how the annotations are connected to each other (mainGraph), and which genes belong to which annotation, and which list they originated from (allAnnotation).

5 Visualization

Currently the easiest way to visualize and interact with this data is by using Cytoscape and the `RCytoscape` package. To set up `RCytoscape`, see the `RCytoscape` website.

Once you have Cytoscape up and running with the CytoscapeRPC plugin running, then we can examine the results from each category of annotations. First up, GO Biological Process.

```
> cw.BP <- ccOutCyt(ccResults$BP,ccOpts)

[1] "shape"
[1] "Desc"
[1] "listMembership"
[1] "compIndx"
[1] "fillcolor"
[1] "toolTip"
[1] "isSig"
```

```
[1] "T10"
[1] "T48"
[1] "T10.Pvalue"
[1] "T10.FDR"
[1] "T10.OddsRatio"
[1] "T10.ExpCount"
[1] "T10.Count"
[1] "T10.Size"
[1] "T48.Pvalue"
[1] "T48.FDR"
[1] "T48.OddsRatio"
[1] "T48.ExpCount"
[1] "T48.Count"
[1] "T48.Size"
[1] "label"
[1] "weight"
```

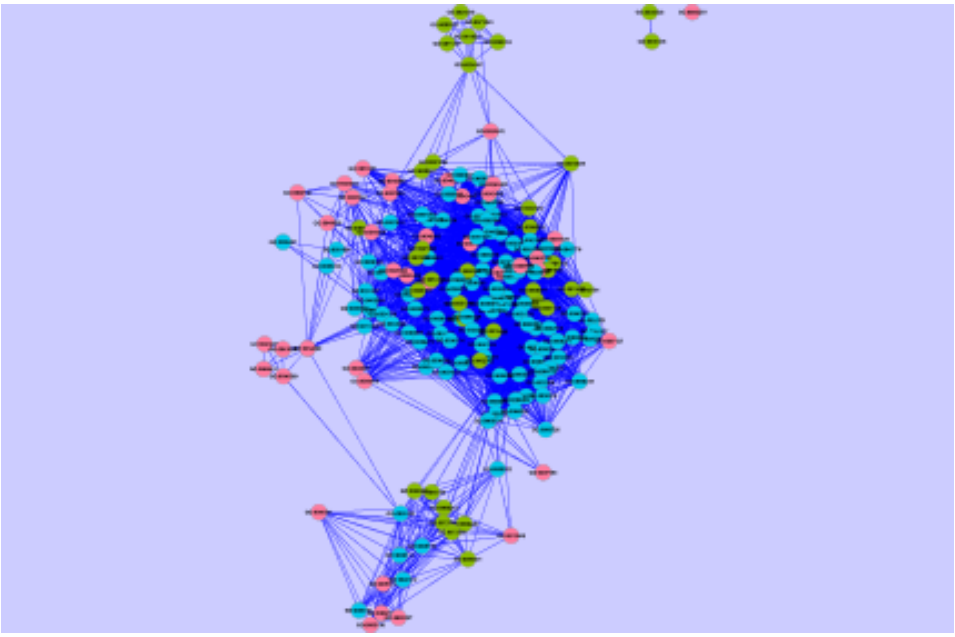

Figure 1: Cytoscape output of the GO BP results

You should now see something in Cytoscape that somewhat resembles the Fig. 1. Reddish nodes came from T10, green from T48, and the blue ones from both. The edges determine that some of the genes are shared between annotations (nodes), and are weighted by how many genes are shared. The graph is layed out using a 'force-directed' layout, and the force on the edges is determined by the number of shared genes. Right now there are a few groupings of nodes, that are probably functionally related. However, there is also a large mass of interconnected nodes in the middle, due to the shared genes in the annotation. We may get a better picture

of this if we "break" the edges between nodes that share lower numbers of genes. This layout is based on the work of Bader and co-workers

```
> breakEdges(cw.BP,0.2)
> breakEdges(cw.BP,0.4)
> breakEdges(cw.BP,0.6)
> breakEdges(cw.BP,0.8)
```

```
[1] TRUE
```

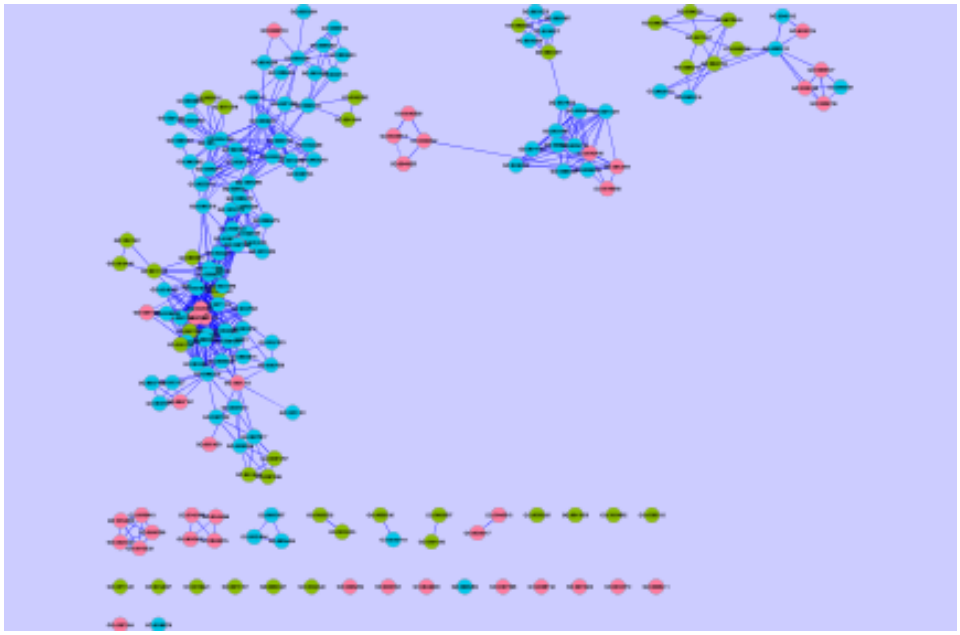

Figure 2: After breaking edges having a weight less than 0.8

By the time you get to breaking any edges with a weight less than 0.8, you should see some very distinct groups of nodes (see Fig. 2). Because the numbers of genes shared between these nodes is high, it is likely that these groups of nodes describe a functional "theme" that will hopefully tell you something about the genes involved in the biological process that you are examining. This view also shows that even if there are no nodes that explicitly show up in both lists, if there are a series of annotations from each list in a well connected group, then perhaps there is still some similarity between the lists in this area.

To see a description of the nodes and their listMembership, as well as other information about each node, you can use the "Data Panel" in Cytoscape, and select the node attributes that you want listed when you select a node. To discover the "theme" of a group of nodes, select all the nodes that belong to a group.

To view the GO nodes in the GO directed-acyclic graph (DAG) hierarchy, we need to change the graph type and re-run `ccCompare` function. The output is shown in Fig. 3.

```
> graphType(enrichLists$BP) <- "hierarchical"
> ccResultsBPHier <- ccCompare(enrichLists$BP, ccOpts)
```

```
> ccResultsBPHier

Annotation category:  GO   BP
Main graph: A graphNEL graph with directed edges
Number of Nodes = 421
Number of Edges = 811

> cw.BP2 <- ccOutCyt(ccResultsBPHier, ccOpts, "BPHier")

[1] "shape"
[1] "Desc"
[1] "listMembership"
[1] "compIndx"
[1] "fillcolor"
[1] "toolTip"
[1] "isSig"
[1] "T10"
[1] "T48"
[1] "T10.Pvalue"
[1] "T10.FDR"
[1] "T10.OddsRatio"
[1] "T10.ExpCount"
[1] "T10.Count"
[1] "T10.Size"
[1] "T48.Pvalue"
[1] "T48.FDR"
[1] "T48.OddsRatio"
[1] "T48.ExpCount"
[1] "T48.Count"
[1] "T48.Size"
[1] "label"
[1] "weight"

[1] TRUE
```

Note that the current hierarchical layout in Cytoscape does not seem to generate layouts that are easy to follow. This layout should only be used when there are small numbers of GO annotations.

We can do a similar process for the KEGG pathways as well (Fig. 4).

```
> cw.KEGG <- ccOutCyt(ccResults$KEGG,ccOpts)

[1] "shape"
[1] "Desc"
[1] "listMembership"
[1] "compIndx"
```

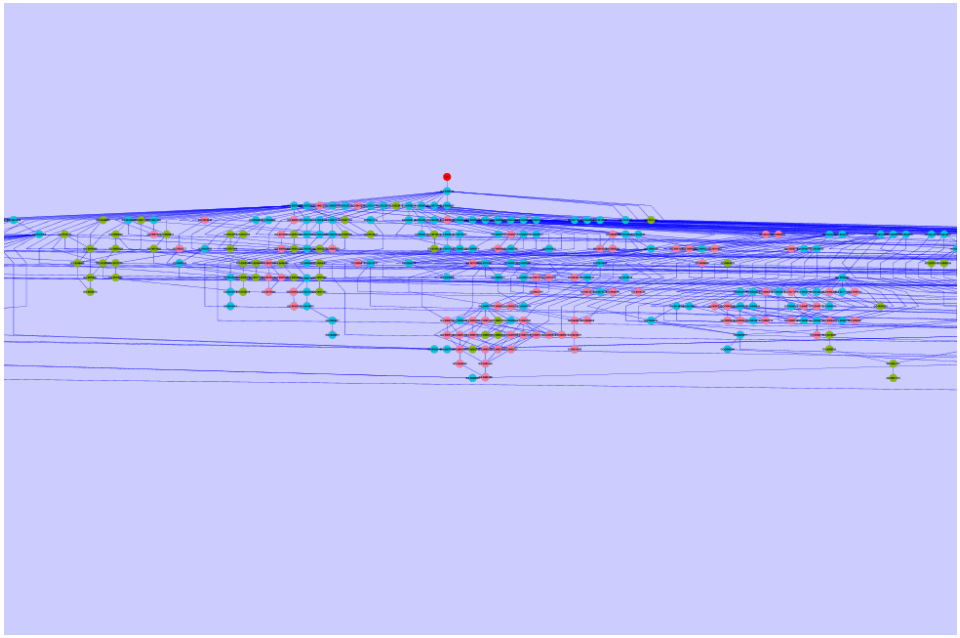

Figure 3: GO BP results using the hierarchical layout

```
[1] "fillcolor"
[1] "toolTip"
[1] "isSig"
[1] "T10"
[1] "T48"
[1] "T10.Pvalue"
[1] "T10.FDR"
[1] "T10.OddsRatio"
[1] "T10.ExpCount"
[1] "T10.Count"
[1] "T10.Size"
[1] "T48.Pvalue"
[1] "T48.FDR"
[1] "T48.OddsRatio"
[1] "T48.ExpCount"
[1] "T48.Count"
[1] "T48.Size"
[1] "label"
[1] "weight"

[1] TRUE
```

If you don't feel that there are enough nodes to work with the data, you may want to change the P-value cutoff used using `pvalueCutoff`, or even the type of P-value, using `pvalueType`.

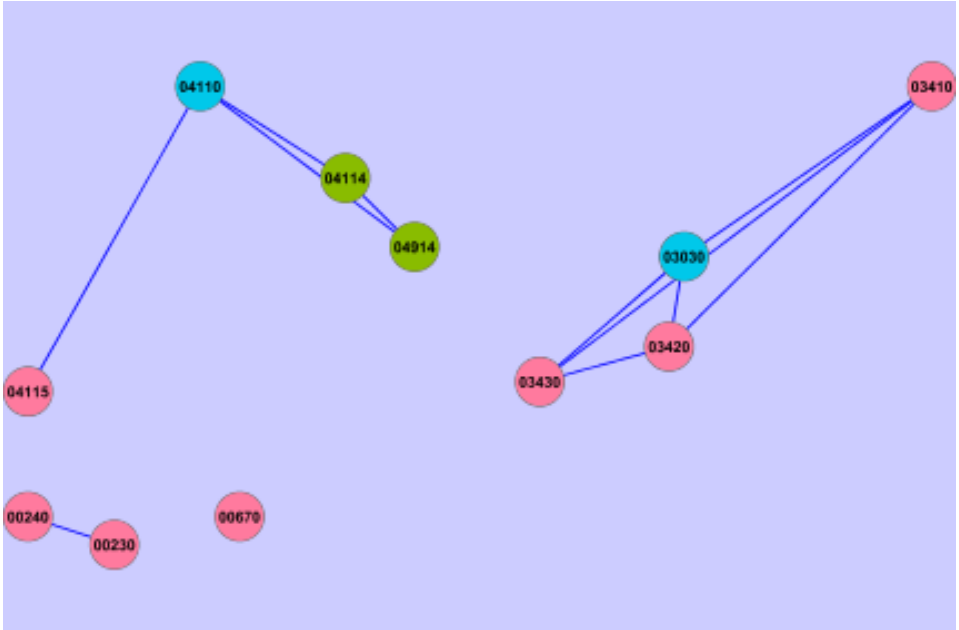

Figure 4: KEGG pathway results

6 Acknowledgements

This package depends heavily on classes and functionality from Category, graph, and the interactive network visualization capabilities enabled by RCytoscape.

7 References

Flight RM, Petruska JC, Harrison BJ, Mohammed F, Rouchka EC "categoryCompare: Meta-analysis of high-throughput experiments using feature annotations", in preparation

```
> sessionInfo()

R version 3.0.2 (2013-09-25)
Platform: x86_64-unknown-linux-gnu (64-bit)

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=en_US.UTF-8        LC_COLLATE=C
 [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

attached base packages:
[1] parallel  stats     graphics  grDevices utils
```

```
[6] datasets  methods   base

other attached packages:
 [1] KEGG.db_2.10.1        GO.db_2.10.1
 [3] categoryCompare_1.6.0 hgu95av2cdf_2.13.0
 [5] limma_3.18.0          estrogen_1.8.10
 [7] genefilter_1.44.0     hgu95av2.db_2.10.1
 [9] org.Hs.eg.db_2.10.1   RSQLite_0.11.4
[11] DBI_0.2-7             AnnotationDbi_1.24.0
[13] affy_1.40.0           Biobase_2.22.0
[15] BiocGenerics_0.8.0

loaded via a namespace (and not attached):
 [1] AnnotationForge_1.4.0 BiocInstaller_1.12.0
 [3] Category_2.28.0       GOstats_2.28.0
 [5] GSEABase_1.24.0       IRanges_1.20.0
 [7] Matrix_1.0-14         RBGL_1.38.0
 [9] RCurl_1.95-4.1        RCytoscape_1.12.0
[11] XML_3.98-1.1          XMLRPC_0.3-0
[13] affyio_1.30.0         annotate_1.40.0
[15] colorspace_1.2-4      graph_1.40.0
[17] grid_3.0.2            hwriter_1.3
[19] lattice_0.20-24       preprocessCore_1.24.0
[21] splines_3.0.2         stats4_3.0.2
[23] survival_2.37-4       tools_3.0.2
[25] xtable_1.7-1          zlibbioc_1.8.0
```