EnrichmentBrowser: Seamless navigation through combined results of set-based and network-based enrichment analysis

Ludwig Geistlinger

February 24, 2015

Contents

1 Introduction

The *EnrichmentBrowser* package implements essential functionality for the enrichment analysis of gene expression data. The analysis combines the advantages of set-based and network-based enrichment analysis in order to derive high-confidence gene sets and biological pathways that are differentially regulated in the expression data under investigation. Besides, the package facilitates the visualization and exploration of such sets and pathways.
To demonstrate the functionality of the package, we consider microarray expression data of patients suffering from acute lymphoblastic leukemia [1]. A frequent chromosomal defect found among these patients is a translocation, in which parts of chromosome 9 and 22 swap places. This results in the oncogenic fusion gene BCR/ABL created by positioning the ABL1 gene on chromosome 9 to a part of the BCR gene on chromosome 22.
We load the *ALL* dataset

```
> library(ALL)
> data(ALL)
```

and select B-cell ALL patients with and without the BCR/ABL fusion as it has been described previously [2].

```
> ind.bs <- grep("^B", ALL$BT)
> ind.mut <- which(ALL$mol.biol %in% c("BCR/ABL", "NEG"))
> sset <- intersect(ind.bs, ind.mut)
> eset <- ALL[, sset]
```

Typically, the expression data is not already available as an *ExpressionSet* in *R* but rather has to be read in from file. This can be done using the function `read.eset`, which reads the expression data (exprs) along with the phenotype data (pdat) and feature data (fdat) into an *ExpressionSet*.

```
> library(EnrichmentBrowser)
> data.dir <- system.file("extdata", package="EnrichmentBrowser")
> exprs.file <- file.path(data.dir, "ALL_exprs.tab")
> pdat.file <- file.path(data.dir, "ALL_pData.tab")
> fdat.file <- file.path(data.dir, "ALL_fData.tab")
> eset2 <- read.eset(exprs.file, pdat.file, fdat.file)
```

We can now access the expression values, which are intensity measurements on a log-scale for 12,625 probes (rows) across 79 patients (columns).

```
> exprs(eset)[1:4,1:4]
```

```
            01005    01010    03002    04007
1000_at   7.597323 7.479445 7.567593 7.905312
1001_at   5.046194 4.932537 4.799294 4.844565
1002_f_at 3.900466 4.208155 3.886169 3.416923
1003_s_at 5.903856 6.169024 5.860459 5.687997
```

```
> dim(exprs(eset))
```

```
[1] 12625    79
```

The phenotype data should contain for each patient a binary group assignment indicating here whether the BCR-ABL gene fusion is present (1) or not (0).

```
> grp <- ifelse(eset$mol.biol == "BCR/ABL", "1", "0")
> pData(eset)$GROUP <- grp
> table(pData(eset)$GROUP)
```

```
 0  1
42 37
```

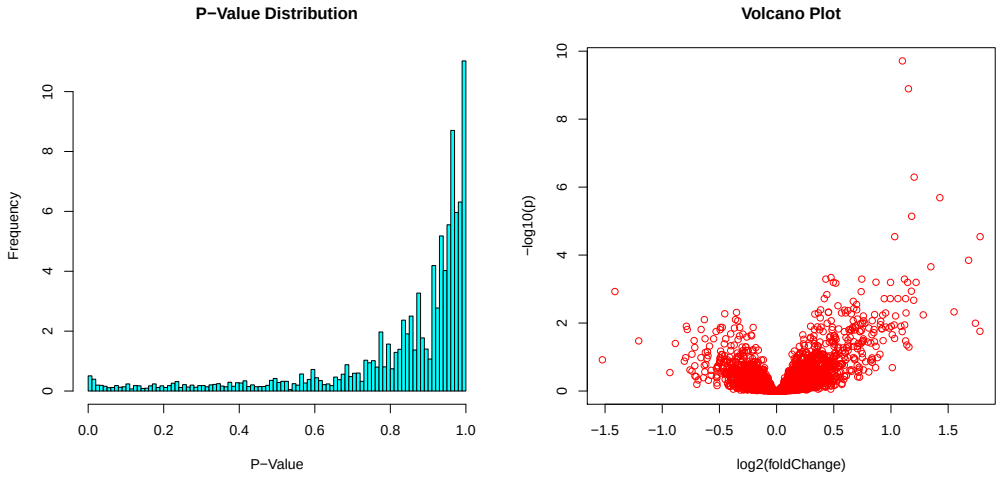
The `de.ana` function carries out a differential expression analysis between the two groups using the function `get.fold.change.and.t.test` from the *simpleaffy* package.
Resulting fold changes and t-test derived p-values for each probe are appended to the `fData` slot.

```
> eset <- de.ana(eset)
> head(fData(eset), n=4)
```

```
                   FC  RAW.PVAL  ADJ.PVAL
1000_at    0.04296986 0.4653192 0.9205936
1001_at    0.03208350 0.6508453 0.9597802
1002_f_at -0.06582929 0.1303694 0.7126578
1003_s_at -0.01270016 0.8307160 0.9811369
```

Raw p-values (`RAW.PVAL`) are already corrected for multiple testing (`ADJ.PVAL`) using the method from Benjamini and Hochberg implemented in the function `p.adjust` from the *stats* package.
To get a first overview, we inspect the p-value distribution and the volcano plot (fold change against p-value).

```
> par(mfrow=c(1,2))
> pdistr(eset)
> volcano(eset)
```



The expression change of highest statistical significance is observed for the probe `1636_g_at`.

```
> fData(eset)[ which.min(fData(eset)$ADJ.PVAL), ]
```

```
                FC     RAW.PVAL     ADJ.PVAL
1636_g_at 1.100012 1.531812e-14 1.933913e-10
```

This turns out to be ABL1 oncogene itself (hsa:25@KEGG). As we often have more than one probe per gene, we compute gene expression values as the average of the corresponding probe values.

```
> gene.eset <- probe.2.gene.eset(eset)
> head(fData(gene.eset))
```

```
              FC   RAW.PVAL  ADJ.PVAL
5595  0.04296986 0.46466091 0.9066708
7075  0.03208350 0.65055402 0.9523681
1557 -0.04394014 0.28966888 0.8226330
643  -0.02775435 0.57200353 0.9342505
1843 -0.42730253 0.08387369 0.5678564
4319 -0.01804432 0.65122308 0.9523681
```

(Note, that the mapping from probe to gene is done automatically as long as as you have the corresponding annotation package, here the *hgu95av2.db* package, installed. Otherwise, the mapping can be defined in the `fData` slot.)

```
> head(fData(eset2))
```

```
              PROBE GENE           FC  RAW.PVAL  ADJ.PVAL
1000_at     1000_at 5595  0.042969860 0.4660552 0.8621742
1010_at     1010_at 5600 -0.095741600 0.1429818 0.6323363
1011_s_at 1011_s_at 7531 -0.184200784 0.2121401 0.7301974
1013_at     1013_at 4090  0.116059597 0.2391914 0.7603127
1018_at     1018_at 7480  0.011548130 0.8393739 0.9662051
1019_g_at 1019_g_at 7480  0.004277009 0.9365001 0.9939091
```

Now, we subject the ALL gene expression data to the enrichment analysis.

2 Set-based enrichment analysis

In the following, we introduce how the *EnrichmentBrowser* package can be used to perform state-of-the-art enrichment analysis of gene sets. We consider the ALL gene expression set as it has been processed in the previous section. We are now interested whether there are not only single genes that are differentially expressed, but also sets of genes known to work together, e.g. as defined by their membership in KEGG pathways.
Hence, we use the function `get.kegg.genesets`, which is based on functionality from the *KEGGREST* package, to download all human KEGG pathways as gene sets.

```
> # hsa.gs <- get.kegg.genesets("hsa")
> gmt.file <- file.path(data.dir, "hsa_kegg_gs.gmt")
> hsa.gs <- parse.genesets.from.GMT(gmt.file)
> length(hsa.gs)
```

```
[1] 39
```

```
> hsa.gs[1:2]
```

```
$hsa05416_Viral_myocarditis
 [1] "100509457" "101060835" "1525"      "1604"      "1605"      "1756"      "1981"
 [8] "1982"      "25"        "2534"      "27"        "3105"      "3106"      "3107"
[15] "3108"      "3109"      "3111"      "3112"      "3113"      "3115"      "3117"
[22] "3118"      "3119"      "3122"      "3123"      "3125"      "3126"      "3127"
[29] "3133"      "3134"      "3135"      "3383"      "3683"      "3689"      "3908"
[36] "4624"      "4625"      "54205"     "5551"      "5879"      "5880"      "5881"
[43] "595"       "60"        "637"       "6442"      "6443"      "6444"      "6445"
[50] "71"        "836"       "841"       "842"       "857"       "8672"      "940"
[57] "941"       "942"       "958"       "959"

$`hsa04622_RIG-I-like_receptor_signaling_pathway`
 [1] "10010"  "1147"   "1432"   "1540"   "1654"   "23586"  "26007"  "29110"  "338376"
[10] "340061" "3439"   "3440"   "3441"   "3442"   "3443"   "3444"   "3445"   "3446"
[19] "3447"   "3448"   "3449"   "3451"   "3452"   "3456"   "3467"   "3551"   "3576"
[28] "3592"   "3593"   "3627"   "3661"   "3665"   "4214"   "4790"   "4792"   "4793"
[37] "5300"   "54941"  "55593"  "5599"   "5600"   "5601"   "5602"   "5603"   "56832"
[46] "57506"  "5970"   "6300"   "64135"  "64343"  "6885"   "7124"   "7186"   "7187"
[55] "7189"   "7706"   "79132"  "79671"  "80143"  "841"    "843"    "8517"   "8717"
[64] "8737"   "8772"   "9140"   "9474"   "9636"   "9641"   "9755"
```

Currently, the following set-based enrichment analysis methods are supported

```
> sbea.methods()
```

```
[1] "ora"   "safe"  "gsea"  "samgs"
```

- ORA: Overrepresentation Analysis (simple and frequently used test based on the hypergeometric distribution [3] for a critical review)
- SAFE: Significance Analysis of Function and Expression (generalization of ORA, includes other test statistics, e.g. Wilcoxon's rank sum, and allows to estimate the significance of gene sets by sample permutation; implemented in the *safe* package)
- GSEA: Gene Set Enrichment Analysis (frequently used and widely accepted, uses a Kolmogorov–Smirnov statistic to test whether the ranks of the p-values of genes in a gene set resemble a uniform distribution [4])
- SAMGS: Significance Analysis of Microarrays on Gene Sets (extending the SAM method for single genes to gene set analysis [5])

For demonstration we perform here a basic ORA choosing a significance level α of 0.05.

```
> sbea.res <- sbea(method="ora", eset=gene.eset, gs=hsa.gs, perm=0, alpha=0.05)
> gs.ranking(sbea.res)
```

```
                                    GENE.SET P.VALUE
1                    hsa05416_Viral_myocarditis  0.0207
2 hsa04622_RIG-I-like_receptor_signaling_pathway  0.0301
```

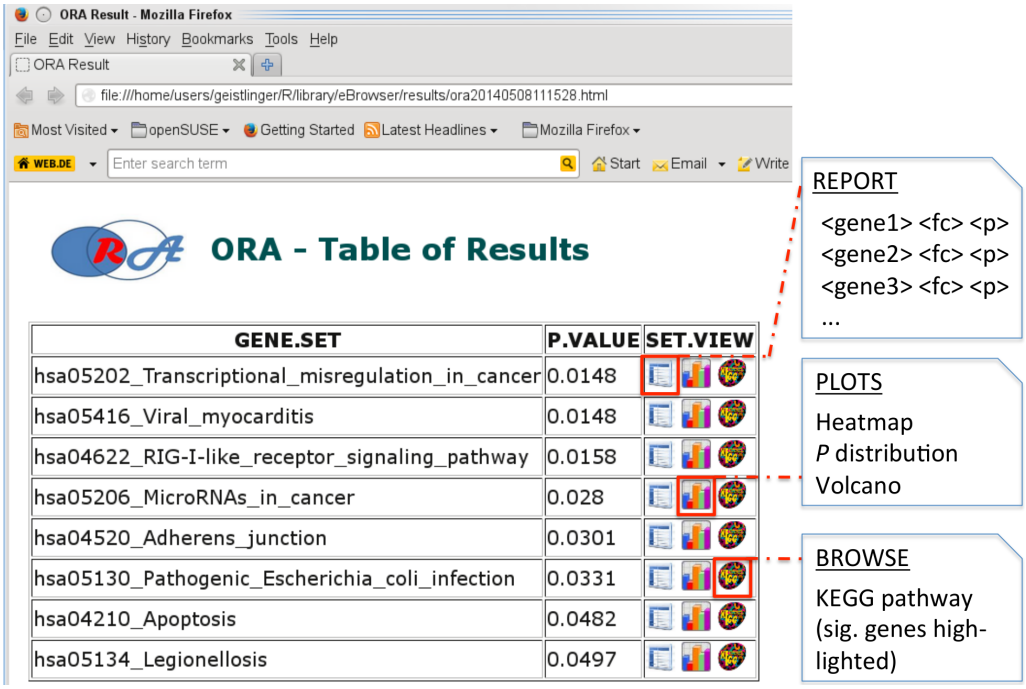


Figure 1: ORA result view. For each significant gene set in the ranking, the user can select to view (1) a basic report, that lists all genes of a set along with fold change and t-test derived p-value, (2) overview plots, such as heatmap, p-value distribution, and volcano plot, (3) the pathway in KEGG with differentially expressed genes highlighted in red.

```
3 hsa05130_Pathogenic_Escherichia_coli_infection  0.0445
4                   hsa00790_Folate_biosynthesis  0.0473
```

The result of every enrichment analysis is a ranking of gene sets by the corresponding p-value. The `gs.ranking` function displays only those gene sets satisfying the chosen significance level α.
While such a ranked list is the standard output of existing enrichment tools, the functionality of the *EnrichmentBrowser* package allows visualization and interactive exploration of resulting gene sets far beyond that point. Using the `ea.browse` function creates a HTML summary from which each gene set can be inspected in more detail. The various options are described in Figure 1.

```
> ea.browse(sbea.res)
```

The goal of the *EnrichmentBrowser* package is to provide the most frequently used enrichment methods. However, it is also possible to exploit its visualization capabilities while using one's own set-based enrichment method. This requires to implement a function that takes the characteristical arguments `eset` (expression data), `gs` (gene sets), `alpha` (significance level), and `perm` (number of permutations). In addition, it must return a numeric vector `ps` storing the resulting p-value for each gene set in `gs`. The p-value vector must be also named accordingly (i.e. `names(ps) == names(gs)`).
Let us consider the following dummy enrichment method, which randomly renders five gene sets significant and all others insignificant.

```
> dummy.sbea <- function(eset, gs, alpha, perm)
+ {
+         sig.ps <- sample(seq(0,0.05, length=1000),5)
+         insig.ps <- sample(seq(0.1,1, length=1000), length(gs)-5)
+         ps <- sample(c(sig.ps, insig.ps), length(gs))
+         names(ps) <- names(gs)
+         return(ps)
+ }
```

We can plug this method into `sbea` as before.

```
> sbea.res2 <- sbea(method="dummy.sbea", eset=gene.eset, gs=hsa.gs)
> gs.ranking(sbea.res2)
                                     GENE.SET P.VALUE
1                          hsa04210_Apoptosis 0.00916
2                 hsa00340_Histidine_metabolism  0.0304
3 hsa04621_NOD-like_receptor_signaling_pathway  0.0362
4                            hsa05144_Malaria  0.0373
5                 hsa03410_Base_excision_repair  0.0462
```

3 Network-based enrichment analysis

Having found sets of genes that are differentially regulated in the ALL data, we are now interested whether these findings can be supported by known regulatory interactions. For example, we want to know whether transcription factors and their target genes are expressed in accordance to the connecting regulations. Such information is usually given in a gene regulatory network derived from specific experiments, e.g. using the *GeneNetworkBuilder*, or compiled from the literature ([6] for an example). There are well-studied processes and organisms for which comprehensive and well-annotated regulatory networks are available, e.g. the `RegulonDB` for *E. coli* and `Yeastract` for *S. cerevisiae*. However, in many cases such a network is missing. A first simple workaround is to compile a network from regulations in the KEGG database.
We can download all KEGG pathways of a specified organism via the `download.kegg.pathways` function that exploits functionality from the *KEGGREST* package.

```
> pwys <- download.kegg.pathways("hsa")
```

In this case, we have already downloaded all human KEGG pathways. We parse them making use of the *KEGGgraph* package and compile the resulting gene regulatory network.

```
> pwys <- file.path(data.dir, "hsa_kegg_pwys.zip")
> hsa.grn <- compile.grn.from.kegg(pwys)
> head(hsa.grn)
```

```
     FROM   TO     TYPE
[1,] "3569" "3570" "+"
[2,] "3458" "3459" "+"
[3,] "3458" "3460" "+"
[4,] "1950" "1956" "+"
[5,] "1950" "2064" "+"
[6,] "1950" "3480" "+"
```

Now we are able to perform enrichment analysis based on the compiled network. Currently the following network-based enrichment analysis methods are supported

```
> nbea.methods()
```

```
[1] "ggea" "nea"  "spia"
```

- GGEA: Gene Graph Enrichment Analysis (evaluates consistency of known regulatory interactions with the observed expression data [7])
- NEA: Network Enrichment Analysis (implemented in the *neaGUI* package)
- SPIA: Signaling Pathway Impact Analysis (implemented in the *SPIA* package)

For demonstration we perform here GGEA using the gene regulatory network compiled above (Note: to produce meaningful p-values of suitable granularity, 1000 permutations is the suggested default. Here, we perform only 100 permutations for demonstration purpose).

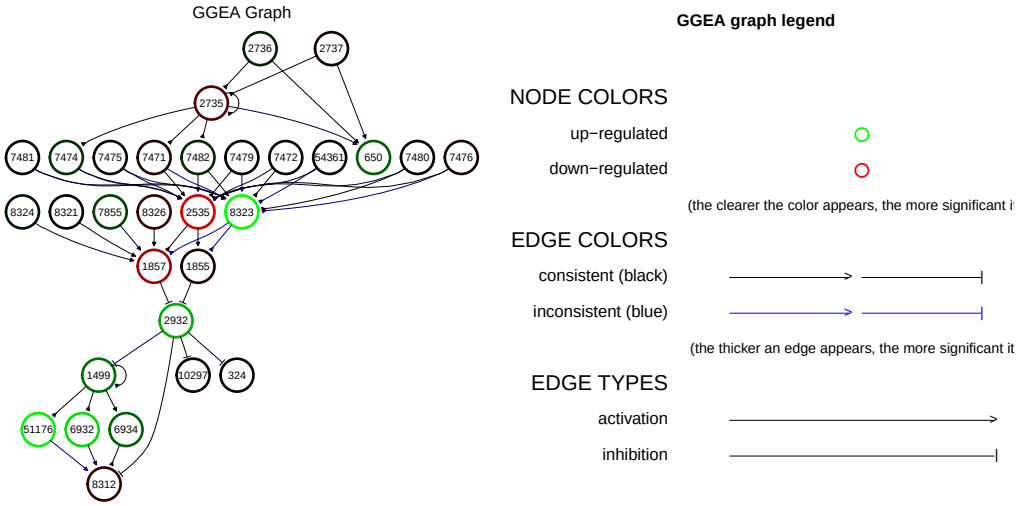
```
> nbea.res <- nbea(method="ggea", eset=gene.eset, gs=hsa.gs, grn=hsa.grn, perm=100)
> gs.ranking(nbea.res)
```

```
                                GENE.SET NR.RELS RAW.SCORE NORM.SCORE P.VALUE
1 hsa05150_Staphylococcus_aureus_infection       8      1.66      0.207       0
2            hsa05323_Rheumatoid_arthritis       8      1.47      0.183       0
3               hsa05416_Viral_myocarditis      23      2.57      0.112       0
4  hsa04514_Cell_adhesion_molecules_(CAMs)      55      4.53     0.0824    0.01
5         hsa04390_Hippo_signaling_pathway     338      8.08     0.0239    0.02
6                         hsa05144_Malaria      13       1.2     0.0927    0.04
```

The resulting ranking lists for each statistically significant gene set the number of relations (NR.RELS) of the given gene regulatory network that involve a gene set member, the sum of consistencies over all relations (RAW.SCORE), the score normalized by induced network size (NORM.SCORE = RAW.SCORE / NR.RELS), and the statistical significance of each gene set based on a permutation approach.
A GGEA graph for a gene set of interest displays the consistency of each interaction in the network that involves a gene set member. Nodes (genes) are colored according to expression (up-/down-regulated) and edges (interactions)

are colored according to consistency, i.e. how well the interaction type (activation/inhibition) is reflected in the correlation of the observed expression of both interaction partners.

```
> par(mfrow=c(1,2))
> ggea.graph(
+     gs=hsa.gs[["hsa05217_Basal_cell_carcinoma"]],
+     grn=hsa.grn, eset=gene.eset)
> ggea.graph.legend()
```



As described in the previous section it is also possible to plug in one's own network-based enrichment method.

4 Combining results

Different enrichment analysis methods usually result in different gene set rankings for the same dataset. To compare results and detect gene sets that are supported by different methods, the *EnrichmentBrowser* package allows to combine results from the different set-based and network-based enrichment analysis methods. The combination of results yields a new ranking of the gene sets under investigation either by the average rank across methods or a combined p-value using Fisher's method or Stouffer's method [8].
We consider the ORA result and the GGEA result from the previous sections and use the function `comb.ea.results`.

```
> res.list <- list(sbea.res, nbea.res)
> comb.res <- comb.ea.results(res.list, pcomb.meth="fisher")
```

The combined result can be detailedly inspected as before and interactively ranked as depicted in Figure 2.

```
> ea.browse(comb.res, graph.view=hsa.grn)
```

eBrowser - Table of Results

GENE.SET	ORA.RANK	GGEA.RANK	AVG.RANK	ORA.PVAL	GGEA.PVAL	COMB.PVAL	SET.VIEW	GRAPH.VIEW
hsa05217_Basal_cell_carcinoma	14	1	8	0.126	0.002	0.00234		
hsa05130_Pathogenic_Escherichia_coli_infection	6	15	10	0.0331	0.096	0.0215		
hsa04622_RIG-I-like_receptor_signaling_pathway	3	26	14	0.0158	0.147	0.0164		
hsa04920_Adipocytokine_signaling_pathway	35	3	19	0.285	0.009	0.0179		
hsa04145_Phagosome	41	4	22	0.327	0.017	0.0344		
hsa00564_Glycerophospholipid_metabolism	29	14	22	0.24	0.093	0.107		
hsa04722_Neurotrophin_signaling_pathway	34	17	26	0.284	0.099	0.129		
hsa05205_Proteoglycans_in_cancer	27	30	28	0.222	0.163	0.156		
hsa05140_Leishmaniasis	38	20	29	0.305	0.111	0.148		
hsa05131_Shigellosis	9	51	30	0.0664	0.308	0.1		
hsa05216_Thyroid_cancer	51	8	30	0.383	0.067	0.12		
hsa00561_Glycerolipid_metabolism	16	43	30	0.132	0.248	0.145		
hsa05416_Viral_myocarditis	2	63	32	0.0148	0.453	0.0403		
hsa05206_MicroRNAs_in_cancer	4	94	49	0.028	0.796	0.107		
hsa04520_Adherens_junction	5	123	64	0.0301	0.908	0.126		
hsa00230_Purine_metabolism	128	5	66	1	0.018	0.0903		
hsa05202_Transcriptional_misregulation_in_cancer	1	146	74	0.0148	0.957	0.0745		
hsa04713_Circadian_entrainment	164	6	85	1	0.021	0.102		
hsa04728_Dopaminergic_synapse	170	7	88	1	0.023	0.11		
hsa05340_Primary_immunodeficiency	218	2	110	1	0.006	0.0367		

Figure 2: Combined result view. By clicking on one of the blue columns (ORA.RANK, ..., COMB.PVAL) the result can be interactively ranked according to the selected criterion.

5 Putting it all together

There are cases where it is necessary to perform some steps of the demonstrated enrichment analysis pipeline individually. However, often it is more convenient to run the complete standardized pipeline. This can be done using the all-in-one wrapper function `ebrowser`. Thus, in order to produce the result page displayed in Figure 2 from scratch, without going through the individual steps listed above, the following call would do the job.

```
> ebrowser(   meth=c("ora", "ggea"),
+         exprs=exprs.file, pdat=pdat.file, fdat=fdat.file,
+         gs=hsa.gs, grn=hsa.grn, comb=TRUE)
```

References

[1] Chiaretti S, Li X, Gentleman R, Vitale A, Vignetti M, and et al. Gene expression profile of adult t-cell acute lymphocytic leukemia identifies distinct subsets of patients with different response to therapy and survival. *Blood*, 103(7):2771–8, 2004.

[2] Gentleman R, Carey V, Huber W, Irizarry R, and Dudoit S. Bioinformatics and computational biology solutions using r and bioconductor. *Springer*, New York, 2005.

[3] Goeman JJ and Buehlmann P. Analyzing gene expression data in terms of gene sets: methodological issues. *Bioinformatics*, 23(8):980–7, 2007.

[4] Subramanian A, Tamayo P, Mootha VK, Mukherjee S, Ebert BL, and et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc Natl Acad Sci USA*, 102(43):15545–50, 2005.

[5] Dinu I, Potter JD, Mueller T, Liu Q, Adewale AJ, and et al. Improving gene set analysis of microarray data by sam-gs. *BMC Bioinformatics*, 8:242, 2007.

[6] Geistlinger L, Csaba G, Dirmeier S, Kueffner R, and Zimmer R. A comprehensive gene regulatory network for the diauxic shift in saccharomyces cerevisiae. *Nucleic Acids Res*, 41(18):8452–63, 2013.

[7] Geistlinger L, Csaba G, Kueffner R, Mulder N, and Zimmer R. From sets to graphs: towards a realistic enrichment analysis of transcriptomic systems. *Bioinformatics*, 27(13):i366–73, 2011.

[8] Kim SC, Lee SJ, Lee WJ, Yum YN, and Kim JH et al. Stouffer's test in a large scale simultaneous hypothesis testing. *PLoS One*, 8(5):e63290, 2013.