GSVA - The Gene Set Variation Analysis Package

Sonja Hänzelmann, Robert Castelo and Justin Guinney

May 7, 2011

Abstract

The GSVA package implements a non-parametric unsupervised method, called Gene Set Variation Analysis (GSVA), for assessing gene set enrichment (GSE) in gene expression microarray data. In contrast to most GSE methods, GSVA performs a change in coordinate systems, transforming the data from a gene by sample matrix to a gene set by sample matrix. Thereby allowing for the evaluation of pathway enrichment for each sample. This transformation is done without the use of a phenotype, thus facilitating very powerful and open-ended analyses in a now pathway centric manner. In this vignette we illustrate how to use the GSVA package to perform some of these analyses using published microarray data already pre-processed and stored in the companion experimental data package GSVAdata.

1 Introduction

Gene set enrichment analysis (GSEA) (see Mootha et al., 2003; Subramanian et al., 2005) is a method designed to assess the concerted behavior of functionally related genes forming a set, between two well-defined groups of samples. Because it does not rely on a "gene list" of interest but on the entire ranking of genes, GSEA has been shown to provide greater sensitivity to find gene expression changes of small magnitude that operate coordinately in specific sets of functionally related genes. However, due to the reduced costs in genome-wide gene-expression assays, data is being produced under more complex experimental designs that involve multiple RNA sources enriched with a wide spectrum of phenotypic and/or clinical information. The Cancer Genome Atlas (TCGA) project (see `http://cancergenome.nih.gov`) and the data deposited on it constitute a canonical example of this situation.

To facilitate the functional enrichment analysis of this kind of data, we developed Gene Set Variation Analysis (GSVA) which allows one to assess the underlying pathway activity variation by transforming the gene by sample matrix into a gene-set by sample matrix without the *a priori* knowledge of the experimental design. The method is both non-parametric and unsupervised, and bypasses the conventional approach of explicitly modeling phenotypes within enrichment scoring algorithms. Focus is therefore placed on the *relative* enrichment of pathways across the sample space rather than the *absolute* enrichment with respect to a single phenotype. The value of this approach is that it permits the use of traditional analytical methods such as classification, survival, clustering, and correlation analysis in a pathway focused manner. It also facilitates sample-wise comparisons between pathways and other complex data types such as microRNA expression or binding data, copy-number variation (CNV) data, or single nucleotide polymorphisms (SNPs). However, for case-control or single phenotype experiments, or where the sample size is not sufficiently large (< 30), other GSE methods that explicitly include the phenotype in their model are more likely to provide greater statistical power to detect functional enrichment.

In the rest of this vignette we describe briefly the methodology behind GSVA, give an overview of the functions implemented in the package and show a few applications. The interested reader is referred to (Hänzelmann et al., 2011) for more comprehensive explanations and more complete data analysis examples with GSVA, as well as for citing GSVA if you use it in your own work.

2 GSVA enrichment scores

GSVA enrichment scores are calculated from two main inputs: a matrix $X = \{x_{ij}\}_{p \times n}$ of expression values for p genes through n samples, where typically $p \gg n$, and a collection of gene sets $\Gamma = \{\gamma_1, \ldots, \gamma_m\}$.

We shall denote by x_i the expression profile of the i-th gene, by x_{ij} the specific expression value of the i-th gene in the j-th sample, and by γ_k the subset of row indices in X such that $\gamma_k \subset \{1, \dots p\}$ defines a set of genes forming a pathway or some other functional unit. Let $|\gamma_k|$ be the number of genes in the gene set.

The first step in the calculation consists of evaluating whether a gene i is highly or lowly expressed in sample j in the context of the sample population distribution, with an expression-level statistic calculated as follows. First, for each gene expression profile $x_i = \{x_{i1}, \dots, x_{in}\}$, a non-parametric kernel estimation of its cumulative density function is performed using a Gaussian kernel (Silverman, 1986, pg. 148):

$$\hat{F}_{h_i}(x_{ij}) = \frac{1}{n} \sum_{k=1}^{n} \int_{-\infty}^{\frac{x_{ij}-x_{ik}}{h_i}} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} \, dt \,. \tag{1}$$

where h_i is the gene-specific bandwidth parameter that controls the resolution of the kernel estimation and is taken as $h_i = s_i/4$, where s_i is the sample standard deviation of the i-th gene. Second, the expression-level statistic is calculated as logarithm of the relative likelihood, or odds, that the i-th gene is expressed in sample j:

$$z_i j = \log \left(\frac{\hat{F}_{h_i}(x_{ij})}{1 - \hat{F}_{h_i}(x_{ij})} \right) . \tag{2}$$

The clearest context for differential gene expression arises under a bi-modal distribution of the data. Here, z_{ij}'s corresponding to the lower mode will have a large negative statistic, higher modes will have a large positive statistic, and samples between the two modes will have z_{ij}'s at or near zero. Genes with uniform or unimodal population distributions do not preclude large negative or positive expression-level statistics, i.e. x_{ij} lying in the tail of a distribution. To dampen the effect of potential outliers, the expression-level statistics z_{ij} are first converted to a rank statistic $z_{(i)j} = p/2 - \text{rank}_j(z_{ij})$ for each j-th sample, before evaluating the enrichment scores.

We assess the enrichment score similarity to the GSEA method Subramanian et al. (2005) using a Kolmogorov-Smirnov (K-S) like random walk statistic:

$$\nu_{jk}(\ell) = \frac{\sum_{i=1}^{\ell} |z_{(i)j}|^{\tau} I(g_{(i)} \in \gamma_k)}{\sum_{i=1}^{p} |z_{(i)j}|^{\tau} I(g_{(i)} \in \gamma_k)} - \frac{\sum_{i=1}^{\ell} I(g_{(i)} \notin \gamma_k)}{p - |\gamma_k|}, \tag{3}$$

where τ is a parameter describing the weight of the tail in the random walk (default $\tau = 1$), γ_k is the k-th gene set, $I(g_{(i)} \in \gamma_k)$ is the indicator function on whether the i-th gene (the gene corresponding to the i-th ranked expression-level statistic) is in gene set γ_k, $|\gamma_k|$ is the number of genes in the k-th gene set, and p is the number of genes in the data set.

Two approaches are possible for turning the K-S random walk statistic into an enrichment score (ES). The first approach is the previously described maximum deviation method Subramanian et al. (2005) where the ES for the j-th sample with respect to the k-th gene set is the maximum deviation of the random walk from zero:

$$ES_j^k = \nu_{jk}[\arg \max_{\ell=1,..,p} |\nu_{jk}(\ell)|]. \tag{4}$$

For each gene set k, this approach produces a distribution of enrichment scores that is bi-modal. Within the supervised paradigm, a "normalized" enrichment score could be obtained via a permutation of the phenotypic labels; since we are operating without labels, we propose a second approach that produces an ES distribution that is approximately *normal*:

$$ES_j^k = \max(0, \nu_{jk}(1), \dots, \nu_{jk}(p)) + \min(0, \nu_{jk}(1), \dots, \nu_{jk}(p)) \tag{5}$$

This approach takes the magnitude difference between the largest positive and negative random walk deviations, and has the effect of dampening out large enrichment scores if there is both a large positive and negative deviation in the random walk. For analyses that require an enrichment score distribution approximately normal, we recommend this alternative method.

3 Overview of the package

The GSVA package implements the methodology described in the previous section in the function `gsva()` which requires two main input arguments: the gene expression data and a collection of gene sets. The expression data can be provided either as a *matrix* object of genes (rows) by sample (columns) expression values, or as an *ExpressionSet* object. The collection of gene sets can be provided either as a *list* object with names identifying gene sets and each entry of the list containing the gene identifiers of the genes forming the corresponding set, or as a *GeneSetCollection* object as defined in the GSEABase package.

When the two main arguments are an *ExpressionSet* object and a *GeneSetCollection* object, the `gsva()` function will first filter out from the gene sets those identifiers that do not match to the chip annotation associated to the *ExpressionSet* object through the function `mapIdentifiers()` from the GSEABase package. This means that both input arguments may specify features with different types of identifiers, like Entrez IDs and probeset IDs, and the GSEABase package will take care to map them to one another. After this first filtering step, it will perform again a second one on the gene sets where those identifiers that do not match to the feature names in the *ExpressionSet* object will be also discarded. If the expression data is given as a *matrix* object then only the latter filtering step will be taken by the `gsva()` function and, therefore, it will be the responsibility of the user to have the same type of identifiers in both the expression data and the gene sets.

After these automatic filtering steps, we may additionally filter out gene sets that do not meet a minimum and/or maximum size specified through the optional arguments `min.sz` and `max.sz` which are set by default to 1 and `Inf`, respectively. Finally, the `gsva()` function will carry out the calculations specified in the previous section and return a gene-set by sample matrix of GSVA enrichment scores in the form of a *matrix* object, if this was the class of the input expression data object or, otherwise, it will return an *ExpressionSet* object inheriting all the corresponding phenotypic information from the input data.

An important argument of the `gsva()` function is the flag `mx.diff` which is set to `TRUE` by default. Under this default setting, GSVA enrichment scores are calculated using Equation 5 and therefore, more amenable by analysis techniques that assume the data to be normally distributed. When setting `mx.diff=FALSE`, then Equation 4 is employed, calculating enrichment in an analogous way to classical GSEA which typically provides bi-modal distribution of GSVA enrichment scores for each gene.

Since the calculations for each gene set are independent from each other, the `gsva()` function offers two possibilities to perform them in parallel. One consists of loading the library snow, which will enable the parallelization of the calculations through a cluster of computers. In order to activate this option we should specify in the argument `parallel.sz` the number of processors we want to use (default is zero which means no parallelization is going to be employed). The other is possibility is loading the library multicore and then the `gsva()` function will use the core processors of the computer where R is running. If we want to limit `gsva()` in the number of core processors that is should use we can do it by specifying such a value in the `parallel.sz` argument.

The other two functions of the GSVA package are `filterGeneSets()` and `computeGeneSetOverlaps()` that serve to explicitly filter out gene sets by size and by pairwise overlap, respectively. Note that the size filter can be also applied within the `gsva()` function call.

4 Applications

In this section we illustrate the following applications of GSVA:

- Functional enrichment between two subtypes of leukemia.
- Identification of molecular signatures in distinct glioblastoma subtypes.
- Meta-pathway analysis in the leukemia data.

Throughout this vignette we will use the C2 collection of literature curated gene sets that form part of the Molecular Signatures Database (MSigDB) version 3.0. This particular collection of gene sets is provided as a *GeneSetCollection* object called `c2BroadSets` in the companion experimental data package GSVAdata, which stores these and other data employed in this vignette. These data can be loaded as follows:

```
> library(GSEABase)
> library(GSVAdata)
> data(c2BroadSets)
> c2BroadSets
```

where we observe that `c2BroadSets` contains 3272 gene sets. We also need to load the following additional libraries:

```
> library(Biobase)
> library(genefilter)
> library(limma)
> library(RColorBrewer)
> library(graph)
> library(Rgraphviz)
> library(GSVA)
```

As a final setup step for this vignette, we will employ the `cache()` function from the Biobase package in order to load some pre-computed results and speed up the building time of the vignette:

```
> cacheDir <- system.file("extdata", package = "GSVA")
> cachePrefix <- "cache4vignette_"
```

In order to enforce re-calculating everything, either the call to the `cache()` function should be replaced by its first argument, or the following command should be written in the R console at this point:

```
> file.remove(paste(cacheDir, list.files(cacheDir,
+     pattern = cachePrefix), sep = "/"))
```

4.1 Functional enrichment

In this section we illustrate how to identify functionally enriched gene sets between two phenotypes. As in most of the applications one starts by calculating GSVA enrichment scores and afterwards, in this case, we will employ the linear modeling techniques implemented in the limma package to find the enriched gene sets.

The data set we are going to use in this section corresponds to the microarray data from (Armstrong et al., 2002) which consists of 37 different individuals with human acute leukemias, where 20 of them had conventional childhood acute lymphoblastic leukemia (ALL) and the other 17 were affected with the MLL (mixed-lineage leukemia gene) translocation. This leukemia data set is stored as an `ExpressionSet` object called `leukemia` in the GSVAdata package and details on how the data was pre-processed can be found on its help page. Enclosed with the RMA expression values we can find some metadata including the main phenotype corresponding to the leukemia sample subtype.

```
> data(leukemia)
> leukemia_eset
```

```
ExpressionSet (storageMode: lockedEnvironment)
assayData: 12626 features, 37 samples
  element names: exprs
protocolData
  sampleNames: CL2001011101AA.CEL CL2001011102AA.CEL
    ... CL2001011152AA.CEL (37 total)
  varLabels: ScanDate
  varMetadata: labelDescription
phenoData
  sampleNames: CL2001011101AA.CEL CL2001011102AA.CEL
    ... CL2001011152AA.CEL (37 total)
  varLabels: subtype
  varMetadata: labelDescription channel
```

```
featureData: none
experimentData: use 'experimentData(object)'
Annotation: hgu95a
```

```
> head(pData(leukemia_eset))
```

```
                   subtype
CL2001011101AA.CEL     ALL
CL2001011102AA.CEL     ALL
CL2001011104AA.CEL     ALL
CL2001011105AA.CEL     ALL
CL2001011109AA.CEL     ALL
CL2001011110AA.CEL     ALL
```

```
> table(leukemia_eset$subtype)
```

```
ALL MLL
 20  17
```

Let's examine the variability of the expression profiles across samples by plotting the cumulative distribution of IQR values as shown in Figure 1. About 50% of the probesets show very limited variability across samples and, therefore, in the following non-specific filtering step we will filter out this fraction from further analysis.

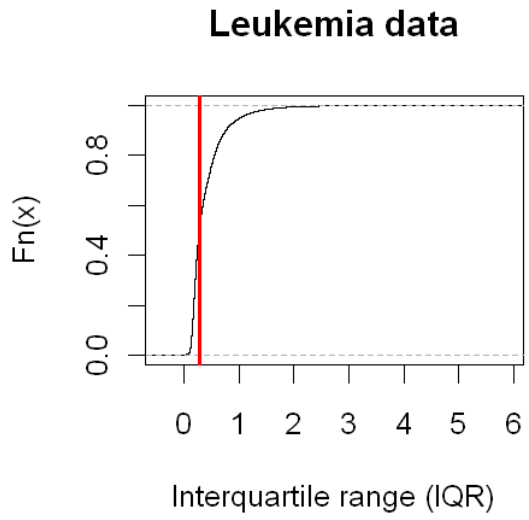

Figure 1: Empirical cumulative distribution of the interquartile range (IQR) of expression values in the leukemia data. The vertical red bar is located at the 50% quantile value of the cumulative distribution.

We carry out a non-specific filtering step by discarding the 50% of the probesets with smaller variability, probesets without Entrez ID annotation, probesets whose associated Entrez ID is duplicated in the annotation, and Affymetrix quality control probes:

```
> filtered_eset <- nsFilter(leukemia_eset, require.entrez = TRUE,
+     remove.dupEntrez = TRUE, var.func = IQR, var.filter = TRUE,
+     var.cutoff = 0.5, filterByQuantile = TRUE,
+     feature.exclude = "^AFFX")
> filtered_eset
```

```
$eset
ExpressionSet (storageMode: lockedEnvironment)
assayData: 4408 features, 37 samples
  element names: exprs
protocolData
  sampleNames: CL2001011101AA.CEL CL2001011102AA.CEL
    ... CL2001011152AA.CEL (37 total)
  varLabels: ScanDate
  varMetadata: labelDescription
phenoData
  sampleNames: CL2001011101AA.CEL CL2001011102AA.CEL
    ... CL2001011152AA.CEL (37 total)
  varLabels: subtype
  varMetadata: labelDescription channel
featureData: none
experimentData: use 'experimentData(object)'
Annotation: hgu95a

$filter.log
$filter.log$numDupsRemoved
[1] 2933

$filter.log$numLowVar
[1] 4409

$filter.log$numRemoved.ENTREZID
[1] 857

$filter.log$feature.exclude
[1] 19
```

```
> leukemia_filtered_eset <- filtered_eset$eset
```

The calculation of GSVA enrichment scores is performed in one single call to the `gsva()` function. However, one should take into account that this function performs further non-specific filtering steps prior to the actual calculations in order to, in one hand, match gene identifiers between gene sets and gene expression values and, on the other hand, meet minimum and maximum gene-set size requirements specified with the arguments `min.sz` and `max.sz`, respectively, which, in the call below, are set to 10 and 500 genes. Because we want to use limma on the resulting GSVA enrichment scores, we let the argument `mx.diff` to its default `TRUE` value.

```
> cache(leukemia_es <- gsva(leukemia_filtered_eset,
+     c2BroadSets, min.sz = 10, max.sz = 500, verbose = FALSE)$es.obs,
+     dir = cacheDir, prefix = cachePrefix)
```

Here we show how to employ GSVA to identify gene sets that are differentially activated for a single dichotomous phenotype. We use the MSigDB C2 version 3.0 database of gene sets Subramanian et al. (2005) of curated pathways. We test whether there is a difference between the GSVA enrichment scores from each pair of phenotypes using a simple linear model and moderated t-statistics computed by the `limma` package using an empirical Bayes shrinkage method (see Smyth, 2004). We are going to examine both, changes at gene level and changes at pathway level and since, as we shall see below, there are plenty of them, we are going to employ the following stringent cutoffs to attain a high level of statistical and biological significance:

```
> adjPvalueCutoff <- 0.001
> logFCcutoff <- log2(2)
```

where we will use the latter only for the gene-level differential expression analysis.

```
> design <- model.matrix(~factor(leukemia_es$subtype))
> colnames(design) <- c("ALL", "MLLvsALL")
> fit <- lmFit(leukemia_es, design)
> fit <- eBayes(fit)
> allGeneSets <- topTable(fit, coef = "MLLvsALL",
+     number = Inf)
> DEgeneSets <- topTable(fit, coef = "MLLvsALL",
+     number = Inf, p.value = adjPvalueCutoff, adjust = "BH")
> res <- decideTests(fit, p.value = adjPvalueCutoff)
> summary(res)
```

```
    ALL MLLvsALL
-1    2        7
0  2027     2006
1     4       20
```

Thus, there are 27 MSigDB C2 curated pathways that are differentially activated between MLL and ALL at 0.1% FDR. When we carry out the corresponding differential expression analysis at gene level:

```
> logFCcutoff <- log2(2)
> design <- model.matrix(~factor(leukemia_eset$subtype))
> colnames(design) <- c("ALL", "MLLvsALL")
> fit <- lmFit(leukemia_filtered_eset, design)
> fit <- eBayes(fit)
> allGenes <- topTable(fit, coef = "MLLvsALL", number = Inf)
> DEgenes <- topTable(fit, coef = "MLLvsALL", number = Inf,
+     p.value = adjPvalueCutoff, adjust = "BH",
+     lfc = logFCcutoff)
> res <- decideTests(fit, p.value = adjPvalueCutoff,
+     lfc = logFCcutoff)
> summary(res)
```

```
    ALL MLLvsALL
-1    0       72
0     0     4281
1  4408       55
```

Here, 127 genes show up as being differentially expressed with a minimum fold-change of 2 at 0.1% FDR. These overall numbers of genes and pathways that change are better seen through the corresponding volcano plots shown in Figure 2.

The signatures of both, the differentially activated pathways reported by the GSVA analysis and of the differentially expressed genes are shown in Figures 3 and 4, respectively. The gene sets and pathways reported in Figure 3 include many directly related to the ALL and MLL leukemias and, among the rest, we could highlight, for instance, the Lysosome gene set since lysosomal enzyme abnormalities have been reported to be involved in leukemias (Besley et al., 1983).

4.2 Molecular signature identification

In Verhaak et al. (2010) four subtypes of Glioblastoma multiforme (GBM) - proneural, classical, neural and mesenchymal - were identified by the characterization of distinct gene-level expression patterns. Using eight gene-set signatures specific to brain cell types - astrocytes, oligodendrocytes, neurons and cultured astroglial cells - derived from murine models by Cahoy et al. (2008), we replicate the analysis of Verhaak et al. (2010) by employing GSVA to transform the gene expression measurements into enrichment scores for these eight gene sets, without taking the sample subtype grouping into account. We start by loading and giving a first glance to the data, which forms part of the `GSVAdata` package:

```
> data(gbm_VerhaakEtAl)
> gbm_eset
```

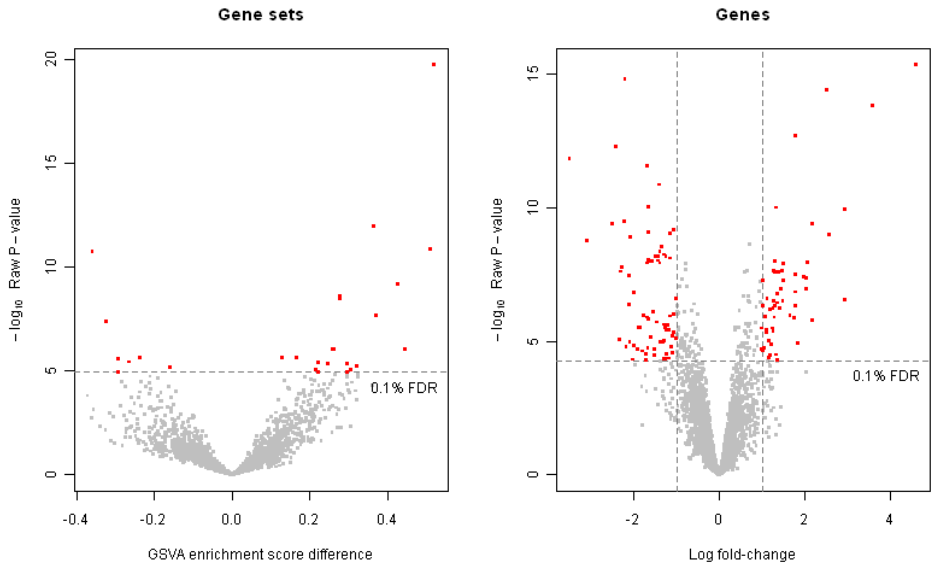

Figure 2: Volcano plots for differential pathway activation (left) and differential gene expression (right) in the leukemia data set.

```
ExpressionSet (storageMode: lockedEnvironment)
assayData: 11861 features, 173 samples
  element names: exprs
protocolData: none
phenoData
  rowNames: TCGA.02.0003.01A.01 TCGA.02.0010.01A.01
    ... TCGA.12.0620.01A.01 (173 total)
  varLabels: subtype
  varMetadata: labelDescription channel
featureData: none
experimentData: use 'experimentData(object)'
Annotation:
```

```
> head(featureNames(gbm_eset))
```

```
[1] "AACS"    "FSTL1"   "ELMO2"   "CREB3L1" "RPS11"
[6] "PNMA1"
```

```
> table(gbm_eset$subtype)
```

```
  Classical Mesenchymal      Neural   Proneural
         38          56          26          53
```

```
> data(brainTxDbSets)
> sapply(brainTxDbSets, length)
```

```
      astrocytic_up       astrocytic_dn        astroglia_up
                 85                  15                  88
       astroglia_dn         neuronal_up         neuronal_dn
                 12                  98                  30
oligodendrocytic_up oligodendrocytic_dn
                 70                  30
```

```
> lapply(brainTxDbSets, head)
```

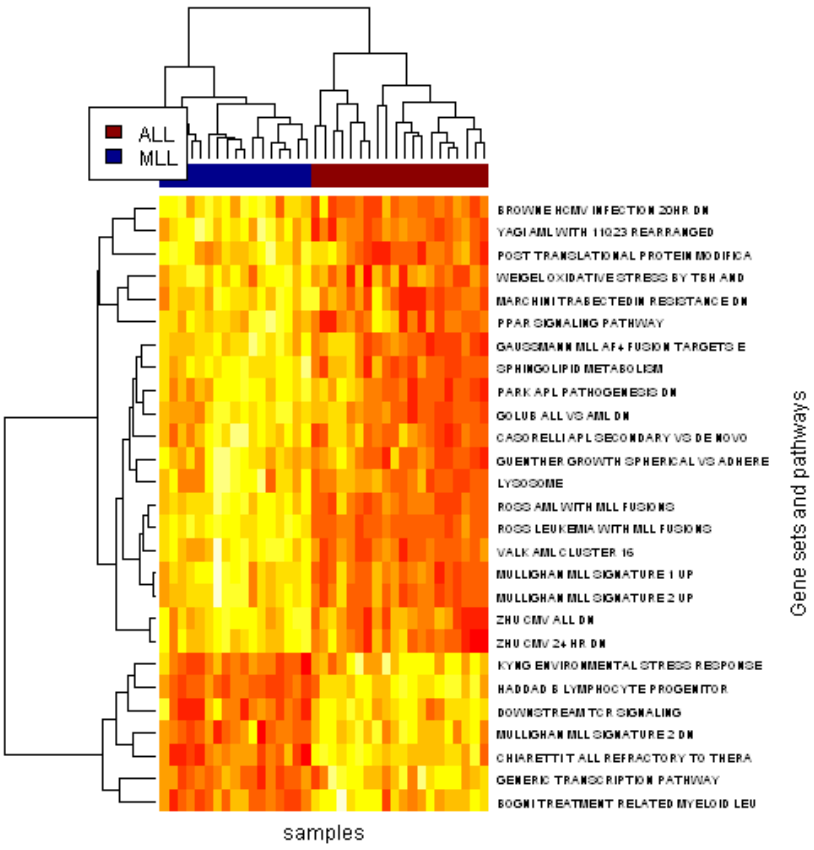

Figure 3: Heatmap of differentially activated pathways at 0.1% FDR in the Leukemia data set.

```
$astrocytic_up
[1] "GRHL1"   "GPAM"    "PAPSS2"  "MERTK"   "BTG1"
[6] "SLC46A1"

$astrocytic_dn
[1] "NPAL3"  "ATP1A1" "FRMD5"  "ASNS"   "SEMA3E" "LPGAT1"

$astroglia_up
[1] "BST2"     "SERPING1" "ACTA2"    "C9orf167" "C1orf31"
[6] "ANXA4"

$astroglia_dn
[1] "PCDH8"   "ATP8A1"  "PHACTR3" "PCDH17"  "CCDC28B"
[6] "TDG"

$neuronal_up
[1] "STXBP1"  "JPH4"    "CACNG3"  "BRUNOL6" "CLSTN2"
[6] "FAM123C"

$neuronal_dn
[1] "DKK3"    "LPHN2"   "AHR"     "NRP1"    "MAP3K15"
[6] "GALNTL4"

$oligodendrocytic_up
[1] "DCT"    "ZNF536" "GNG8"   "ELOVL6" "NR2C1"  "RCBTB1"
```

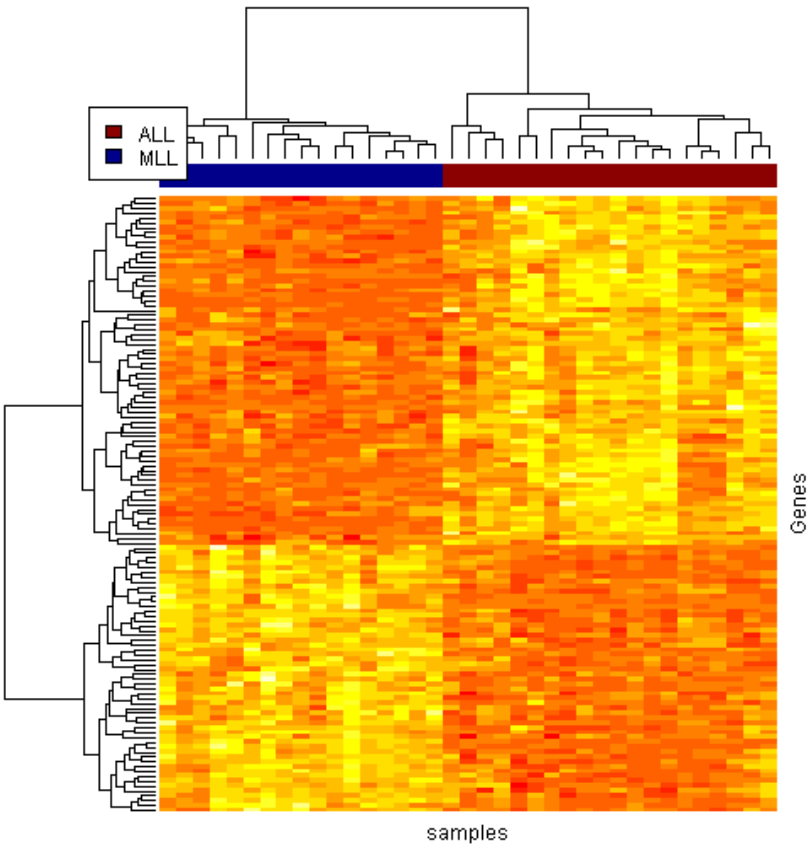

Figure 4: Heatmap of differentially expressed genes with a minimum fold-change of 2 at 0.1% FDR in the leukemia data set.

```
$oligodendrocytic_dn
[1] "DKK3"    "LPHN2"   "AHR"     "NRP1"    "MAP3K15"
[6] "GALNTL4"
```

GSVA enrichment scores for the gene sets contained in `brainTxDbSets` are calculated, in this case using `mx.diff=FALSE`, as follows:

```
> gbm_es <- gsva(gbm_eset, brainTxDbSets, mx.diff = FALSE,
+     verbose = FALSE)$es.obs
```

Figure 5 shows the GSVA enrichment scores obtained for the up-regulated gene sets across the samples of the four GBM subtypes. As expected, the *neural* class is associated with the neural gene set and the astrocytic gene sets. The *mesenchymal* subtype is characterized by the expression of mesenchymal and microglial markers, thus we expect it to correlate with the astroglial gene set. The *proneural* subtype shows high expression of oligodendrocytic development genes, thus it is not surprising that the oligodendrocytic gene set is highly enriched for ths group. Interestingly, the *classical* group correlates highly with the astrocytic gene set. In summary, the resulting GSVA enrichment scores recapitulate accurately the molecular signatures from Verhaak et al. (2010).

4.3 Meta-pathway analysis

In biological systems, pathways do not operate independently and can have diverse degrees of cross-talk between them. We call here a meta-pathway analysis the identification of pathways that have a highly-coordinated activity but whose gene sets have little or no overlap.

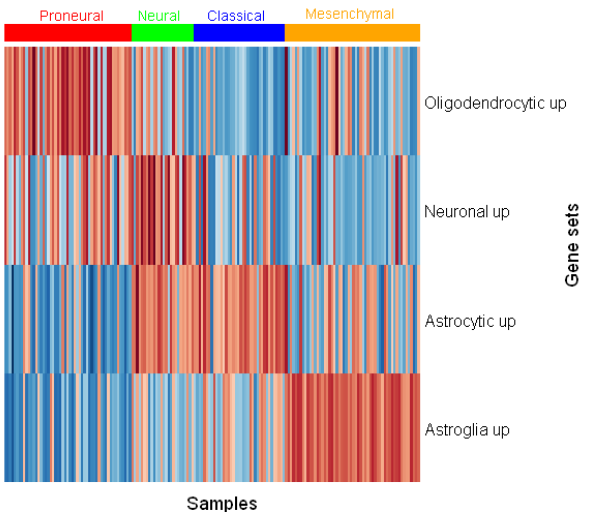

Figure 5: Heatmap of GSVA scores for cell-type brain signatures from murine models (y-axis) across GBM samples grouped by GBM subtype.

For the purpose of simplifying calculations in this vignette, we consider only a subset of the C2 MSigDB Gene Sets, concretely those belonging to the KEGG pathways:

```
> KEGGc2BroadSets <- c2BroadSets[grep("^KEGG", names(c2BroadSets))]
> KEGGc2BroadSets

GeneSetCollection
  names: KEGG_GLYCOLYSIS_GLUCONEOGENESIS, KEGG_CITRATE_CYCLE_TCA_CYCLE, ..., KEGG_VIRAL_MYOCARDITIS
  unique identifiers: 55902, 2645, ..., 1981 (5267 total)
  types in collection:
    geneIdType: EntrezIdentifier (1 total)
    collectionType: BroadCollection (1 total)
```

We calculate GSVA enrichment scores discarding gene sets with less than 10 genes and more than 500. Note that we do not filter for variability here as we are not searching of differential pathway activation and we do not do it either for probeset annotations since this step is taken when mapping probesets to gene sets:

```
> leukemiaKEGG_es <- gsva(leukemia_eset, KEGGc2BroadSets,
+     min.sz = 10, max.sz = 500, mx.diff = TRUE,
+     verbose = FALSE)$es.obs
```

We are interested in those pathways that have little overlap between their sets of genes but are highly correlated. For the purpose of applying such a filter we need to calculate the fraction of genes that overlap between every pair of gene sets which is possible to do through the function `computeGeneSetsOverlap()`:

```
> overlapMatrix <- computeGeneSetsOverlap(KEGGc2BroadSets,
+     leukemia_eset, min.sz = 10, max.sz = 500)
```

We can quickly obtain a network of cross-talk associations by calculating marginal pairwise correlations, like Pearson correlation coefficients (PCCs), and selecting those pairs of pathways that are highly correlated. Here below we select pathway associations with an absolute PCC $|\rho| > 0.8$ and a maximum gene set overlap of 5%. We can see the selected pairs forming this network in Figure 6.

```
> pcc <- cor(t(exprs(leukemiaKEGG_es)))
> pcc[overlapMatrix > 0.05] <- 0
> pcc[lower.tri(pcc)] <- 0
```

```
> diag(pcc) <- 0
> arrIdxs <- which(abs(pcc) > 0.8, arr.ind = TRUE)
> pccEdges <- data.frame(PWYi = featureNames(leukemiaKEGG_es)[arrIdxs[,
+     1]], PWYj = featureNames(leukemiaKEGG_es)[arrIdxs[,
+     2]], PCC = pcc[arrIdxs])
```

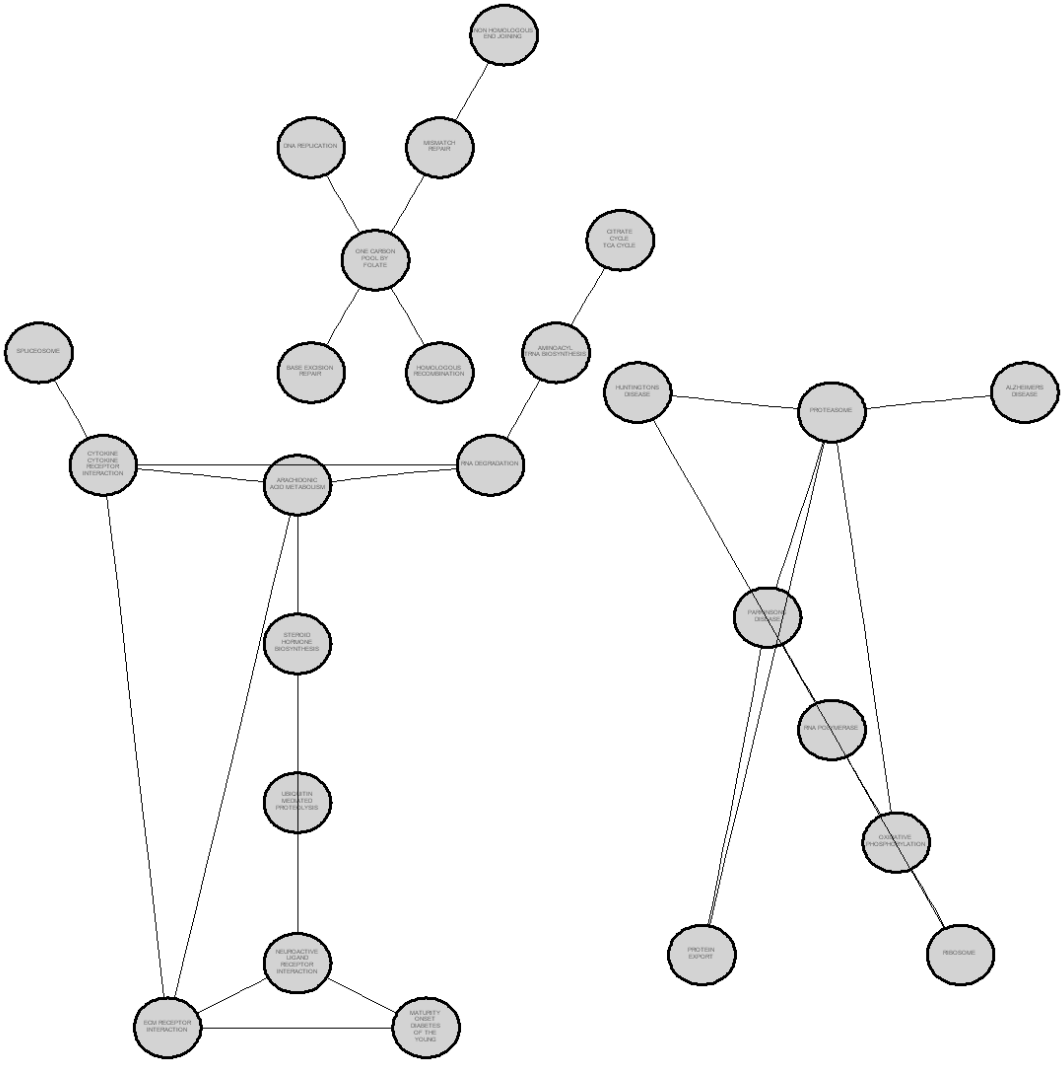

Figure 6: Network of cross-talk associations between KEGG pathways of the leukemia data set obtained by selecting those associations with an absolute value of Pearson correlation $|\rho| > 0.8$ and a maximum gene set overlap of 5%.

Some of the marginal pairwise associations in Figure 6 may be spurious, that is, indirectly mediated by other pathways. In order to select direct (non-spurious) relationships we have carried out a Gaussian graphical modeling (GGM) analysis of the cross-talk associations between pathways that follow from the GSVA enrichment score data. A GGM analysis assumes that the data forms a multivariate normal sample from a distribution $\mathcal{N}(\mu, \Sigma)$ and that the underlying network can be represented by an undirected graph G whose missing edges match the pattern of zeroes in the inverse covariance matrix Σ^{-1} (see Lauritzen, 1996). For this purpose we will employ the qpgraph library:

```
> library(qpgraph)
```

Since the dimension of the data with p =180 and n =37 precludes the application of classical GGM techniques (Lauritzen, 1996, pg. 126) we will follow a limited-order correlation based approach described

in (Castelo and Roverato, 2006, 2009). We start by estimating average non-rejection rates (Castelo and Roverato, 2009) which are a measure of linear association over all marginal distributions of size $(q+2) < n$.

```
> cache(avgnrr <- qpAvgNrr(leukemiaKEGG_es, verbose = FALSE),
+     dir = cacheDir, prefix = cachePrefix)
```

We do not consider the associations involving those pairs of pathways whose gene sets overlap by more than 5%:

```
> avgnrr[overlapMatrix > 0.05] <- NA
```

By considering some cutoff on the average non-rejection rate we could directly obtain an estimated q-order partial correlation graph, or qp-graph, denoted by $\hat{G}^{(q)}$ and which would constitute an approximation to the underlying undirected graph G. However, following the model-based strategy proposed in (Castelo and Roverato, 2006), we will first examine the maximum clique size, denoted by $w(G)$ and also known as the clique number of G, of the possible resulting graphs as function of different cutoffs applied to the average non-rejection rate. This can be done by using the function `qpClique()` whose result is displayed in Figure 7

```
> qpclq <- qpClique(avgnrr, N = dim(leukemiaKEGG_es)[2])
```

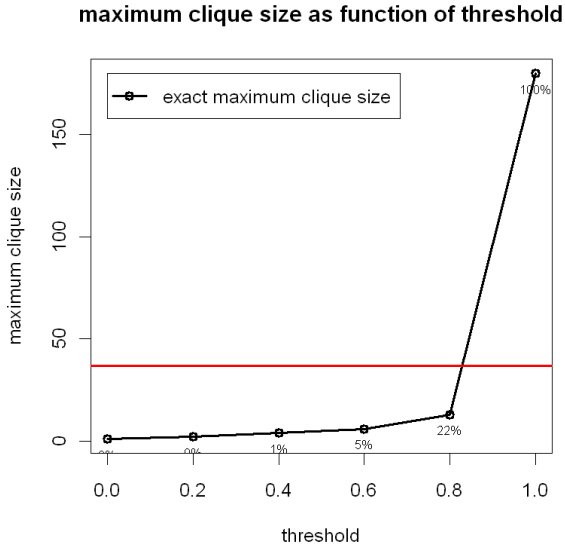

Figure 7: Clique number as function of the average non-rejection rate. The red line indicates the sample size of the leukemia data set (n =37) and next to each point the graph density is indicated.

This function also returns the largest cutoff β^*, among those considered in the plot, whose resulting estimated graph $\hat{G}$ has $w(\hat{G}) < n$. In this case β^* =0.8 and using this cutoff we obtain a resulting qp-graph $\hat{G}^{(q)}$ which has $w(\hat{G}^{(q)})$ =13 as shown here below:

```
> g <- qpGraph(avgnrr, threshold = qpclq$threshold)
> w <- qpCliqueNumber(g, verbose = FALSE)
> w

[1] 13
```

Since $w(\hat{G}^{(q)})$ =13 is smaller than n =37 there is a chance that the maximum likelihood estimate (MLE) of the sample covariance matrix S exists (Lauritzen, 1996, pg. 133), under the restrictions imposed by

the qp-graph $\hat{G}^{(q)}$. Once a MLE of S is obtained, its inverse $\Sigma^{-1} = K = \{\kappa_{ij}\}$ can be calculated and, therefore, the corresponding partial correlation coefficients (PACs) as follows:

$$\rho_{ij.R} = \frac{-\kappa_{ij}}{\sqrt{\kappa_{ii}\,\kappa_{jj}}} \text{ where } R = V\backslash\{i,j\}\,. \tag{6}$$

Since these PACs come from a MLE of the sample covariance matrix S, P-values for the null hypothesis of zero partial correlation can be calculated following (Roverato and Whittaker, 1996). All these computations can be made in one single call to the function `qpPAC()`:

```
> cache(pac <- qpPAC(leukemiaKEGG_es, g, return.K = TRUE,
+     verbose = FALSE), dir = cacheDir, prefix = cachePrefix)
```

We employ the estimated PACs and their P-values to select a final estimate $\hat{G}$ of the underlying undirected graph G whose FWER of including a wrong edge is below a desired network-wide significance level. This control of the FWER helps in discarding spurious associations with a large marginal strength (i.e., a large Pearson correlation) but which in fact are indirectly occurring. In this case we select a $\hat{G}$ with FWER < 0.05 using Holm's procedure as follows:

```
> ridx <- row(pac$P)[as.matrix(upper.tri(pac$P) &
+     g)]
> cidx <- col(pac$P)[as.matrix(upper.tri(pac$P) &
+     g)]
> sigEdges <- which(p.adjust(pac$P[cbind(ridx, cidx)],
+     method = "holm") < 0.05)
> sigEdges <- data.frame(PWYi = colnames(pac$P)[ridx][sigEdges],
+     PWYj = colnames(pac$P)[cidx][sigEdges], PAC = pac$R[cbind(ridx,
+         cidx)][sigEdges], P.value = pac$P[cbind(ridx,
+         cidx)][sigEdges], PCC = cov2cor(solve(pac$K))[cbind(ridx,
+         cidx)][sigEdges])
> sigEdges <- sigEdges[sort(abs(sigEdges$P.value),
+     index.return = TRUE)$ix, ]
> dim(sigEdges)
```

```
[1] 8 5
```

The network shown in Figure 8 contains one large connected component with the KEGG pathway "One carbon pool by folate" (KEGG ID HSA00670) as the most connected node in the network. Folate is an essential nutrient that has been shown to play a role in prevention of many diseases including neural tube defects, cardiovascular disease, and cancer where genetic variation in folate metabolism has been reported to be associated to childhood leukemia (Thompson et al., 2001). The genes that form this pathway, and form part of the analyzed data set, are:

```
> ids <- geneIds(c2BroadSets["KEGG_ONE_CARBON_POOL_BY_FOLATE"])[[1]]
> unlist(mget(ids[!is.na(match(ids, unlist(mget(featureNames(leukemia_eset),
+     hgu95aENTREZID))))], org.Hs.egSYMBOL), use.names = FALSE)
```

```
 [1] "MTHFD2"  "GART"    "TYMS"    "ALDH1L1" "MTHFS"
 [6] "AMT"     "DHFR"    "SHMT1"   "MTR"     "MTHFD1"
[11] "ATIC"    "MTHFR"   "SHMT2"
```

Among these genes, a specific polymorphism in *MTR* has been shown to be associated to an increased risk of ALL which was most pronounced for cases with the MLL translocation (Lightfoot et al., 2010).

5 Session Information

```
> toLatex(sessionInfo())
```

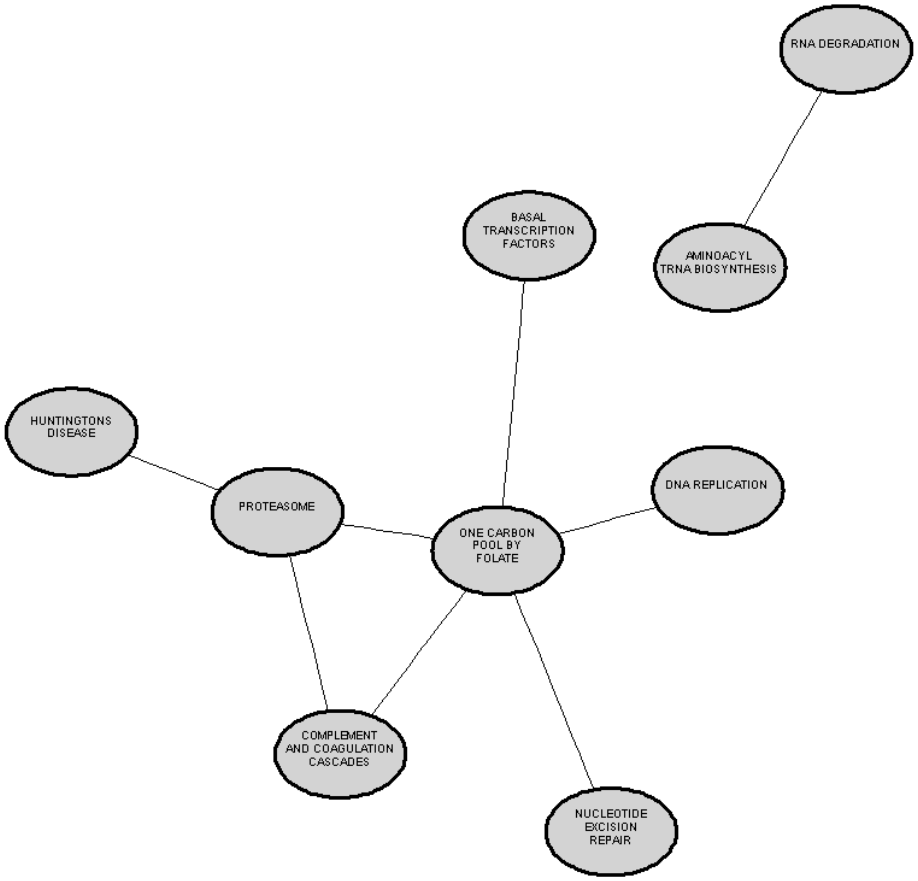

Figure 8: Network of cross-talk associations between KEGG pathways of the leukemia data set obtained by a Gaussian graphical modeling approach by which edges are included in the graph at a FWER $<$ 0.05.

- R version 2.13.0 (2011-04-13), `i386-pc-mingw32`
- Locale: `LC_COLLATE=C`, `LC_CTYPE=English_United States.1252`, `LC_MONETARY=English_United States.1252`, `LC_NUMERIC=C`, `LC_TIME=English_United States.1252`
- Base packages: base, datasets, grDevices, graphics, grid, methods, stats, utils
- Other packages: AnnotationDbi 1.14.1, Biobase 2.12.1, DBI 0.2-5, GSEABase 1.14.0, GSVA 1.0.1, GSVAdata 0.99.0, RColorBrewer 1.0-2, RSQLite 0.9-4, Rgraphviz 1.30.1, annotate 1.30.0, genefilter 1.34.0, graph 1.30.0, hgu95a.db 2.5.0, limma 3.8.1, org.Hs.eg.db 2.5.0, qpgraph 1.8.0
- Loaded via a namespace (and not attached): Matrix 0.999375-50, XML 3.4-0.2, lattice 0.19-26, splines 2.13.0, survival 2.36-9, tools 2.13.0, xtable 1.5-6

References

Armstrong, S. A., Staunton, J. E., Silverman, L. B., Pieters, R., den Boer, M. L., Minden, M. D., Sallan, S. E., Lander, E. S., Golub, T. R., and Korsmeyer, S. J. (2002). MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia. *Nature Gen.*, 30(1):41–7.

Besley, G. T., Moss, S. E., Bain, A. D., and Dewar, A. E. (1983). Correlation of lysosomal enzyme abnormalities in various forms of adult leukaemia. *J Clin Pathol*, 36(9):1000–4.

Cahoy, J. D., Emery, B., Kaushal, A., Foo, L. C., Zamanian, J. L., Christopherson, K. S., Xing, Y., Lubischer, J. L., Krieg, P. A., Krupenko, S. A., Thompson, W. J., and Barres, B. A. (2008). A transcriptome database for astrocytes, neurons, and oligodendrocytes: A new resource for understanding brain development and function. *J. Neurosci.*, 28(1):264–78.

Castelo, R. and Roverato, A. (2006). A robust procedure for gaussian graphical model search from microarray data with p larger than n. *J Mach Learn Res*, 7:2621–2650.

Castelo, R. and Roverato, A. (2009). Reverse engineering molecular regulatory networks from microarray data with qp-graphs. *J Comput Biol*, 16(2):213–27.

Hänzelmann, S., Castelo, R., and Guinney, J. (2011). GSVA: Gene set variation analysis. *submitted*.

Lauritzen, S. (1996). *Graphical models*. Oxford University Press.

Lightfoot, T. J., Johnston, W. T., Painter, D., Simpson, J., Roman, E., Skibola, C. F., Smith, M. T., Allan, J. M., Taylor, G. M., and United Kingdom Childhood Cancer Study (2010). Genetic variation in the folate metabolic pathway and risk of childhood leukemia. *Blood*, 115(19):3923–9.

Mootha, V. K., Lindgren, C. M., Eriksson, K., Subramanian, A., Sihag, S., Lehar, J., Puigserver, P., Carlsson, E., Ridderstrile, M., Laurila, E., Houstis, N., Daly, M. J., Patterson, N., Mesirov, J. P., Golub, T. R., Tamayo, P., Spiegelman, B., Lander, E. S., Hirschhorn, J. N., Altshuler, D., and Groop, L. C. (2003). PGC-1alpha-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. *Nature Genet.*, 34(3):267–73.

Roverato, A. and Whittaker, J. (1996). Standard errors for the parameters of graphical gaussian models. *Stat Comput*, 6:297–302.

Silverman, B. (1986). *Density Estimation for Statistics and Data Analysis*. Chapman and Hall.

Smyth, G. K. (2004). Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Stat Appl Genet Mol Biol*, 3:Article3.

Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., Paulovich, A., Pomeroy, S. L., Golub, T. R., Lander, E. S., and Mesirov, J. P. (2005). Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. U.S.A.*, 102(43):15545–50.

Thompson, J. R., Gerald, P. F., Willoughby, M. L., and Armstrong, B. K. (2001). Maternal folate supplementation in pregnancy and protection against acute lymphoblastic leukaemia in childhood: a case-control study. *Lancet*, 358:1935–1940.

Verhaak, R. G. W., Hoadley, K. A., Purdom, E., Wang, V., Qi, Y., Wilkerson, M. D., Miller, C. R., Ding, L., Golub, T., Mesirov, J. P., Alexe, G., Lawrence, M., O'Kelly, M., Tamayo, P., Weir, B. A., Gabriel, S., Winckler, W., Gupta, S., Jakkula, L., Feiler, H. S., Hodgson, J. G., James, C. D., Sarkaria, J. N., Brennan, C., Kahn, A., Spellman, P. T., Wilson, R. K., Speed, T. P., Gray, J. W., Meyerson, M., Getz, G., Perou, C. M., and Hayes, D. N. (2010). Integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in PDGFRA, IDH1, EGFR, and NF1. *Cancer Cell*, 17(1):98–110.