GSVA: The Gene Set Variation Analysis package for microarray and RNA-seq data

Sonja Hänzelmann[1], Robert Castelo[1] and Justin Guinney[2]

September 21, 2012

1. Research Program on Biomedical Informatics (GRIB), Hospital del Mar Research Institute (IMIM) and Universitat Pompeu Fabra, Parc de Recerca Biomèdica de Barcelona, Doctor Aiguader 88, 08003 Barcelona, Catalonia, Spain

2. Sage Bionetworks, 1100 Fairview Ave N., Seattle, Washington, 98109 USA

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 and RNA-seq 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 and RNA-seq 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 the assessment of 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 phenotype. The value of this approach is that it permits the use of traditional analytical methods such as classification, survival analysis, 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 experiments, or data with a moderate to small sample size (< 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., 2012) 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

A schematic overview of the GSVA method is provided in Figure 1, which shows the two main required inputs: a matrix $X = \{x_{ij}\}_{p \times n}$ of normalized expression values (see Methods for details on the pre-processing steps) for p genes by 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 $\gamma_k \subset \{1, \ldots p\}$ defines a set of genes forming a pathway or some other functional unit. Let $|\gamma_k|$ be the number of genes in γ_k.

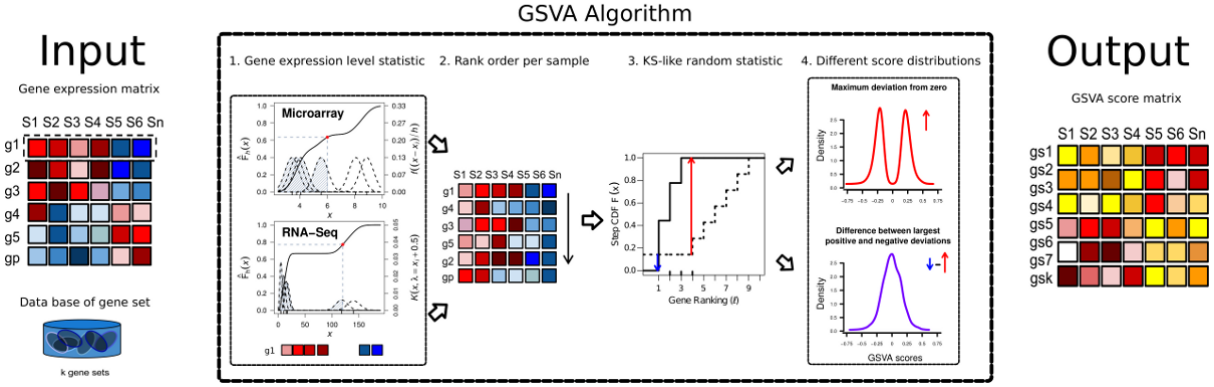


Figure 1: **GSVA methods outline.** The input for the GSVA algorithm are a gene expression matrix in the form of log2 microarray expression values or RNA-seq counts and a database of gene sets. 1. Kernel estimation of the cumulative density function (kcdf). The two plots show two simulated expression profiles mimicking 6 samples from microarray and RNA-seq. The x-axis corresponds to expression values where each gene is lowly expressed in the four samples with lower values and highly expressed in the other two. The scale of the kcdf is on the left y-axis and the scale of the Gaussian and Poisson kernels is on the right y-axis. 2. The expression-level statistic is rank ordered for each sample. 3. For every gene set, the Kolmogorov-Smirnov-like rank statistic is calculated. The plot illustrates a gene set consisting of 3 genes out of a total number of 10 with the sample-wise calculation of genes inside and outside of the gene set. 4. The GSVA enrichment score is either the difference between the two sums or the maximum deviation from zero. The two plots show two simulations of the resulting scores under the null hypothesis of no gene expression change (see main text). The output of the algorithm is matrix containing pathway enrichment profiles for each gene set and sample.

GSVA starts by evaluating whether a gene i is highly or lowly expressed in sample j in the context of the sample population distribution. Probe effects can alter hybridization intensities in microarray data such that expression values can greatly differ between two non-expressed genesZilliox and Irizarry (2007). Analogous gene-specific biases, such as GC content or gene length have been described in RNA-seq dataHansen et al. (2012). To bring distinct expression profiles to a common scale, an expression-level statistic is calculated as follows. For each gene expression profile $x_i = \{x_{i1}, \ldots, x_{in}\}$, a non-parametric kernel estimation of its cumulative density function is performed using a Gaussian kernel (Silverman, 1986, pg. 148) in the case of microarray data:

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

where h_i is the gene-specific bandwidth parameter that controls the resolution of the kernel estimation, which is set to $h_i = s_i/4$, where s_i is the sample standard deviation of the i-th gene (Figure 1, step 1). In the case of RNA-seq data, a discrete Poisson kernel Canale and Dunson (2011) is employed:

$$\hat{F}_r(x_{ij}) = \frac{1}{n} \sum_{k=1}^{n} \sum_{y=0}^{x_{ij}} \frac{e^{-(x_{ik}+r)}(x_{ik}+r)^y}{y!}\,, \tag{2}$$

where $r = 0.5$ in order to set the mode of the Poisson kernel at each x_{ik}, because the mode of a Poisson distribution with an integer mean λ occurs at λ and $\lambda - 1$ and at the largest integer smaller than λ when λ is continuous.

Let z_{ij} denote the previous expression-level statistic $\hat{F}_{h_i}(x_{ij})$, or $\hat{F}_r(x_{ij})$, depending on whether x_{ij} are continuous microarray, or discrete count RNA-seq values, respectively. The following step condenses expression-level statistics into gene sets by calculating sample-wise enrichment scores. To reduce the influence of potential outliers, we first convert z_{ij} to ranks $z_{(i)j}$ for each sample j and normalize further $r_{ij} = |p/2 - z_{(i)j}|$ to make the ranks symmetric around zero (Figure 1, step 2).

We assess the enrichment score similar to the GSEA and ASSESS methods Subramanian et al. (2005); Edelman et al. (2006) using the Kolmogorov-Smirnov (KS) like random walk statistic (Figure 1, step 3):

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

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) belongs to 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.

We offer two approaches for turning the KS like random walk statistic into an enrichment statistic (ES) (also called GSVA score), the classical maximum deviation method Subramanian et al. (2005); Edelman et al. (2006); Verhaak et al. (2010) and a normalized ES. The first ES is the maximum deviation from zero of the random walk of the j-th sample with respect to the k-th gene set :

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

For each gene set k, this approach produces a distribution of enrichment scores that is bimodal (Figure 1, step 4, top panel). This is an intrinsic property of the KS like random walk, which generates non-zero maximum deviations under the null distribution. In GSEA Subramanian et al. (2005) it is also observed that the empirical null distribution obtained by permuting phenotypes is bimodal and, for this reason, significance is determined independently using the positive and negative sides of the null distribution. In our case, we would like to provide a standard Gaussian distribution of enrichment scores under the null hypothesis of no change in pathway activity throughout the sample population. For this purpose we propose a second, alternative score that produces an ES distribution approximating this requirement (Figure 1, step 4, bottom panel):

$$ES_{jk}^{\text{diff}} = \left|ES_{jk}^{+}\right| - \left|ES_{jk}^{-}\right| = \max_{\ell=1,\ldots,p}(0, \nu_{jk}(\ell)) - \min_{\ell=1,\ldots,p}(0, \nu_{jk}(\ell)), \tag{5}$$

where ES_{jk}^{+} and ES_{jk}^{-} are the largest positive and negative random walk deviations from zero, respectively, for sample j and gene set k. This statistic may be compared to the Kuiper test statistic Pearson (1963), which sums the maximum and minimum deviations to make the test statistic more sensitive in the tails. In contrast, our test statistic penalizes deviations that are large in both tails, and provides a "normalization" of the enrichment score by subtracting potential noise. There is a clear biological interpretation of this statistic, it emphasizes genes in pathways that are concordantly activated in one direction only, either over-expressed or under-expressed relative to the overall population. For pathways containing genes strongly acting in both directions, the deviations will cancel each other out and show little or no enrichment. Because this statistic is unimodal and approximately normal, downstream analyses which may impose distributional assumptions on the data are possible.

Figure 1, step 4 shows a simple simulation where standard Gaussian deviates are independenty sampled from $p = 20,000$ genes and $n = 30$ samples, thus mimicking a null distribution of no change in gene expression. One hundred gene sets are uniformly sampled at random from the p genes with sizes ranging from 10 to 100 genes. Using these two inputs, we calculate the maximum deviation ES and the normalized ES. The resulting distributions are depicted in Figure 1, step 4 and in the larger figure below, illustrating the previous description.

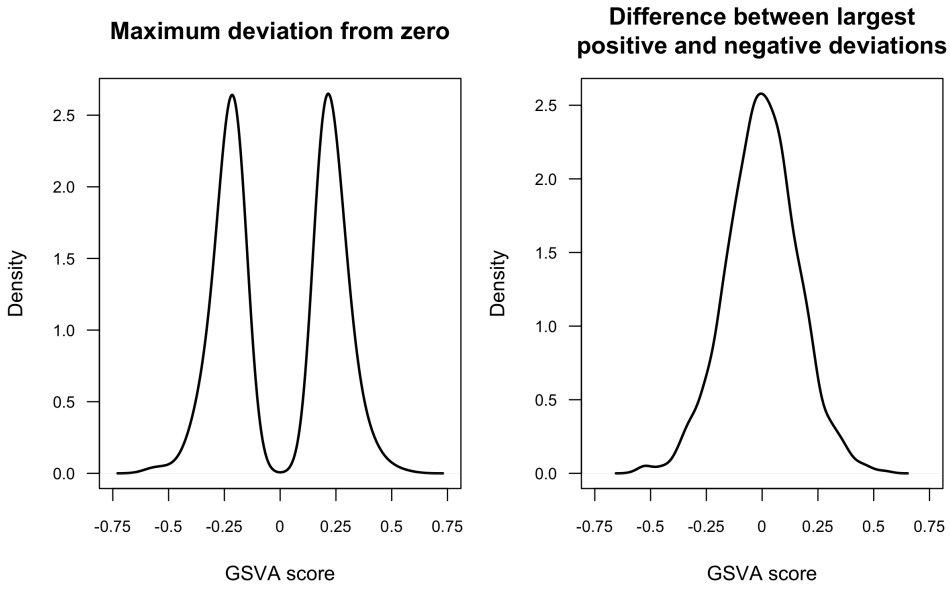
```
> library(GSVA)
> p <- 20000    ## number of genes
```

```
> n <- 30       ## number of samples
> nGS <- 100    ## number of gene sets
> min.sz <- 10  ## minimum gene set size
> max.sz <- 100 ## maximum gene set size
> X <- matrix(rnorm(p*n), nrow=p, dimnames=list(1:p, 1:n))
> dim(X)

[1] 20000    30

> gs <- as.list(sample(min.sz:max.sz, size=nGS, replace=TRUE)) ## sample gene set sizes
> gs <- lapply(gs, function(n, p) sample(1:p, size=n, replace=FALSE), p) ## sample gene sets
> es.max <- gsva(X, gs, mx.diff=FALSE, verbose=FALSE)$es.obs
> es.dif <- gsva(X, gs, mx.diff=TRUE, verbose=FALSE)$es.obs

> par(mfrow=c(1,2), mar=c(4, 4, 4, 1))
> plot(density(as.vector(es.max)), main="Maximum deviation from zero",
+      xlab="GSVA score", lwd=2, las=1, xaxt="n", xlim=c(-0.75, 0.75), cex.axis=0.8)
> axis(1, at=seq(-0.75, 0.75, by=0.25), labels=seq(-0.75, 0.75, by=0.25), cex.axis=0.8)
> plot(density(as.vector(es.dif)), main="Difference between largest\npositive and negative deviation
+      xlab="GSVA score", lwd=2, las=1, xaxt="n", xlim=c(-0.75, 0.75), cex.axis=0.8)
> axis(1, at=seq(-0.75, 0.75, by=0.25), labels=seq(-0.75, 0.75, by=0.25), cex.axis=0.8)
```



Although the GSVA algorithm itself does not evaluate statistical significance for the enrichment of gene sets, significance with respect to one or more phenotypes can be easily evaluated using conventional statistical models. Likewise, false discovery rates can be estimated by permuting the sample labels (Methods). Examples of these techniques are provided in the following section.

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 translate the gene identifiers used in the *GeneSetCollection* object into the corresponding feature identifiers of the *ExpressionSet* object, according to its corresponding annotation package. This translation is carried out by an internal call to the function `mapIdentifiers()` from the GSEABase package. This means that both input arguments may specify features with different types of identifiers, such as Entrez IDs and probeset IDs, and the GSEABase package will take care of mapping them to each other.

A second filtering step is applied that removes genes without matching features in the *ExpressionSet* object. If the expression data is given as a *matrix* object then only the latter filtering step will be applied 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 by 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 when this was the class of the input expression data object. 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, are 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 a bimodal 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 loading the library parallel 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 should be used, we can do it by specifying the number of cores in the `parallel.sz` argument.

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

The `gsva()` function also offers the following three other unsupervised GSE methods that calculate single sample pathway summaries of expression and which can be selected through the `method` argument:

- `method="plage"` (Tomfohr et al., 2005). Pathway level analysis of gene expression (PLAGE) standardizes first expression profiles into z-scores over the samples and then calculates the singular value decomposition $Z_\gamma = UDV'$ on the z-scores of the genes in the gene set. The coefficients of the first right-singular vector (first column of V) are taken as the gene set summaries of expression over the samples.

- `method="zscore"` (Lee et al., 2008). The combined z-score method also, as PLAGE, standardizes first expression profiles into z-scores over the samples, but combines them together for each gene set at each individual sample as follows. Given a gene set $\gamma = \{1, \ldots, k\}$ with z-scores $Z_1, \ldots, Z_k$ for each gene, the combined z-score Z_γ for the gene set γ is defined as:

$$Z_\gamma = \frac{\sum_{i=1}^{k} Z_i}{\sqrt{k}}\,. \tag{6}$$

- `method="ssgsea"` (Barbie et al., 2009). Single sample GSEA (ssGSEA) calculates a gene set enrichment score per sample as the normalized difference in empirical cumulative distribution functions of gene expression ranks inside and outside the gene set.

By default `method="gsva"` and the `gsva()` function uses the GSVA algorithm.

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 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 accompanying 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(RBGL)
> 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 we start by calculating GSVA enrichment scores and afterwards, we will employ the linear modeling techniques implemented in the limma package to find the enriched gene sets.

The data set we use in this section corresponds to the microarray data from (Armstrong et al., 2002) which consists of 37 different individuals with human acute leukemia, where 20 of them have conventional childhood acute lymphoblastic leukemia (ALL) and the other 17 are 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 in the corresponding help page. Enclosed with the RMA expression values we provide 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 2. About 50% of the probesets show very limited variability across samples and, therefore, in the following non-specific filtering step we remove this fraction from further analysis.

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: 4401 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)
```

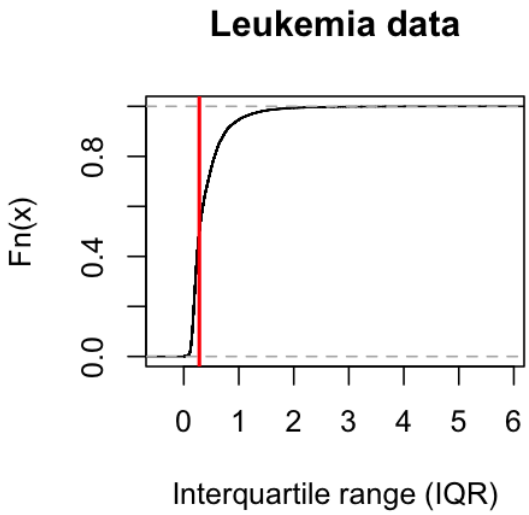


Figure 2: Empirical cumulative distribution of the interquartile range (IQR) of expression values in the leukemia data. The vertical red bar is located at the 50% quantile value of the cumulative distribution.

```
  varLabels: subtype
  varMetadata: labelDescription channel
featureData: none
experimentData: use 'experimentData(object)'
Annotation: hgu95a

$filter.log
$filter.log$numDupsRemoved
[1] 2951

$filter.log$numLowVar
[1] 4401

$filter.log$numRemoved.ENTREZID
[1] 854

$filter.log$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. On the one hand, it matches gene identifiers between gene sets and gene expression values. On the other hand, it discards gene sets that do not 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 leave deliberately unchanged the default argument `mx.diff=TRUE` to obtain approximately normally distributed ES.

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

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 cut-offs 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    3        9
0  2030     1997
1     1       28
```

Thus, there are 37 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       71
0     0     4277
1  4401       53
```

Here, 124 genes show up as being differentially expressed with a minimum fold-change of 2 at 0.1% FDR. We illustrate the genes and pathways that are changing by means of volcano plots (Fig. 3.

The signatures of both, the differentially activated pathways reported by the GSVA analysis and of the differentially expressed genes are shown in Figures 4 and 5, respectively. Many of the gene sets and pathways reported in Figure 4 are directly related to ALL and MLL.

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 have a quick glance to the data which forms part of the `GSVAdata` package:

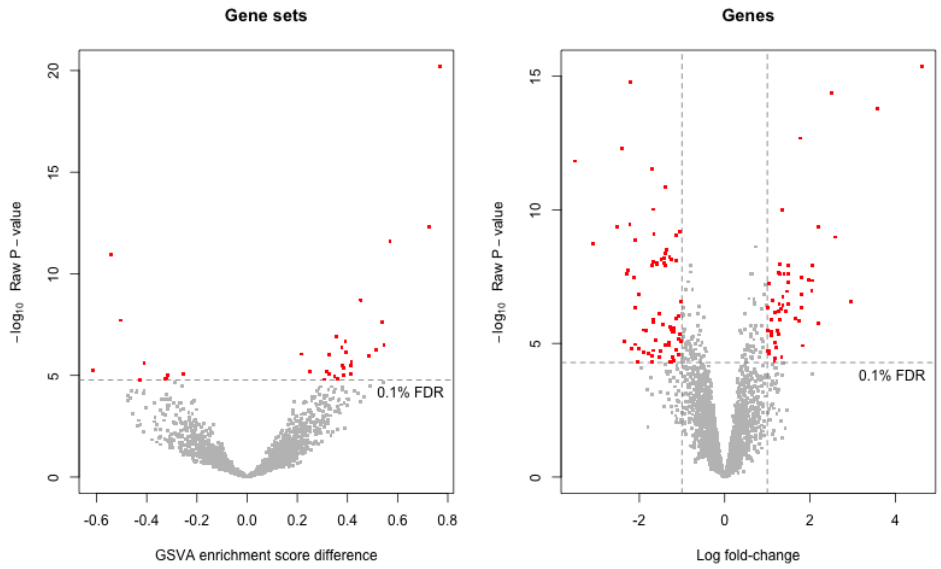


Figure 3: Volcano plots for differential pathway activation (left) and differential gene expression (right) in the leukemia data set.

```
> data(gbm_VerhaakEtAl)
> gbm_eset

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
```

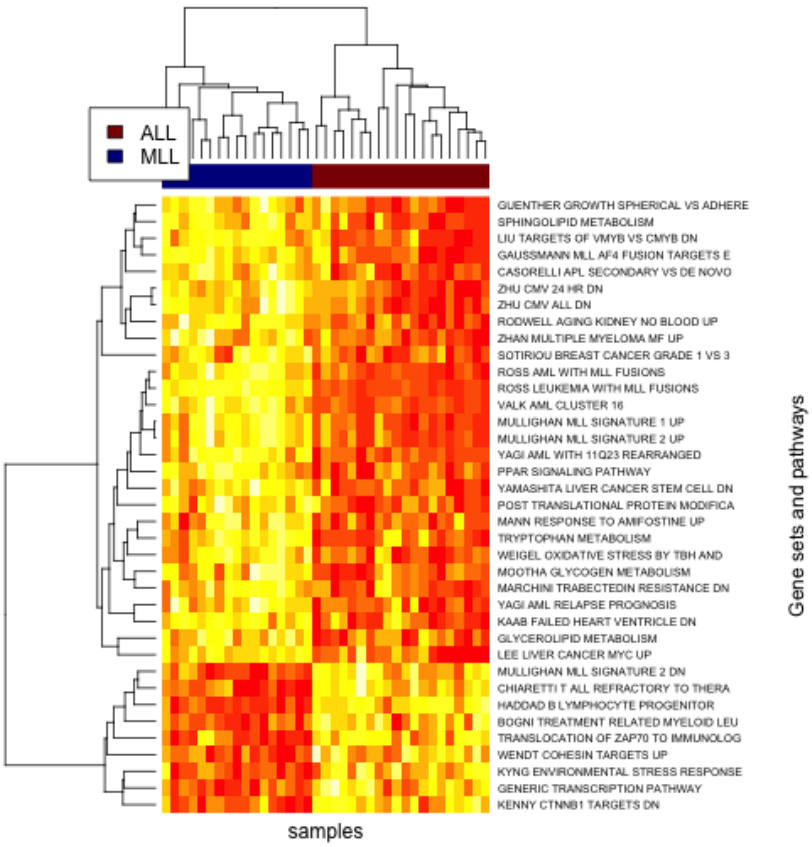


Figure 4: Heatmap of differentially activated pathways at 0.1% FDR in the Leukemia data set.

```
> lapply(brainTxDbSets, head)

$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
```

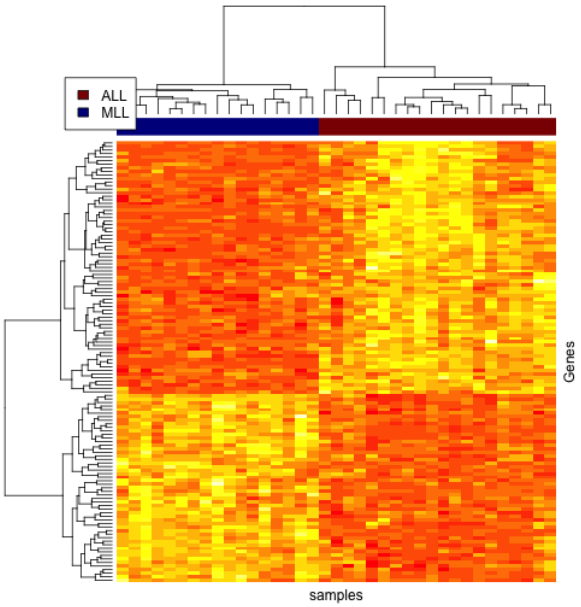


Figure 5: Heatmap of differentially expressed genes with a minimum fold-change of 2 at 0.1% FDR in the leukemia data set.

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

$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 6 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. In this subsection we show how to identify pathways that have a highly-coordinated activity but whose gene sets have little or no overlap. We apply this type of analysis, which we call meta-pathway analysis, to the leukemia data set we previously analyzed for differential pathway activation.

For this analysis we consider the subset of canonical pathways from the C2 collection of MSigDB Gene Sets. These correspond to the following pathways from KEGG, REACTOME and BIOCARTA:

```
> canonicalC2BroadSets <- c2BroadSets[c(grep("^KEGG", names(c2BroadSets)),
+                                       grep("^REACTOME", names(c2BroadSets)),
+                                       grep("^BIOCARTA", names(c2BroadSets)))]
> canonicalC2BroadSets
```

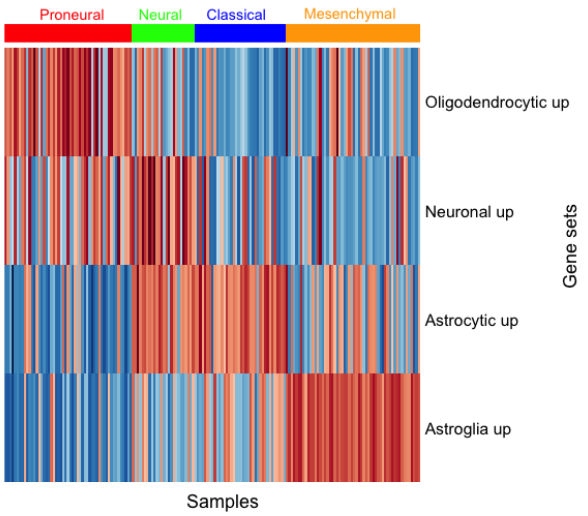


Figure 6: Heatmap of GSVA scores for cell-type brain signatures from murine models (y-axis) across GBM samples grouped by GBM subtype.

```
GeneSetCollection
  names: KEGG_GLYCOLYSIS_GLUCONEOGENESIS, KEGG_CITRATE_CYCLE_TCA_CYCLE, ..., BIOCARTA_ACTINY_PATHWAY
  unique identifiers: 55902, 2645, ..., 8544 (6744 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 as we are not interested for differential pathway activation:

```
> cache(leukemia_canonicalPwy_es <- gsva(leukemia_eset, canonicalC2BroadSets,
+         min.sz=10, max.sz=500, mx.diff=TRUE, verbose=TRUE)$es.obs,
+         dir=cacheDir, prefix=cachePrefix)
```

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 with the function `computeGeneSetsOverlap()`:

```
> overlapMatrix <- computeGeneSetsOverlap(canonicalC2BroadSets, leukemia_eset,
+                                         min.sz=10, max.sz=500)
```

We could quickly obtain a network of cross-talk associations by calculating marginal pair-wise correlations, like Pearson correlation coefficients (PCCs), and selecting those pairs of pathways that are highly correlated. However, potentially many of the marginal pair-wise associations could 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). Since the dimension of the data with p =802 and n =37 precludes the application of classical GGM techniques (Lauritzen, 1996, pg. 126) we will follow a limited-order partial correlation based approach described in (Castelo and Roverato, 2006, 2009) and implemented in the Bioconductor package qpgraph:

```
> library(qpgraph)
```

In this approach one calculates a measure of linear association over all marginal distributions of size $(q+2) < n$, called the non-rejection rate (Castelo and Roverato, 2006). However, the stratification of samples into the two leukemia subtypes ALL and MLL introduces a covariate which could be confounding the estimated linear associations. To take this structure of the data into account, we can employ the so-called generalized non-rejection rate (Roverato and Castelo, 2010), which aims at identifying associations that are common through multiple data sets or experimental conditions. We calculate generalized non-rejection rates using $q = 15$ for ALL and $q = 12$ for MLL, i.e., $q = n - 5$ in each condition:

```
> cache(gennrr <- qpGenNrr(leukemia_canonicalPwy_es, datasetIdx="subtype",
+                          qOrders=c(ALL=15, MLL=12),
+                          identicalQs=FALSE, clusterSize=2)$genNrr,
+       dir=cacheDir, prefix=cachePrefix)
```

We do not consider the associations involving pairs of pathways with an overlap of genes larger than 5%:

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

By considering a cut-off on the generalized non-rejection rate we could directly obtain an estimated q-order partial correlation graph, or qp-graph, denoted by $\hat{G}^{(q)}$, 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 boundary size, denoted by $bd(G)$, of the possible resulting graphs as function of different cut-offs applied to the non-rejection rates. This can be done by using the function `qpBoundary()` whose result is displayed in Figure 7

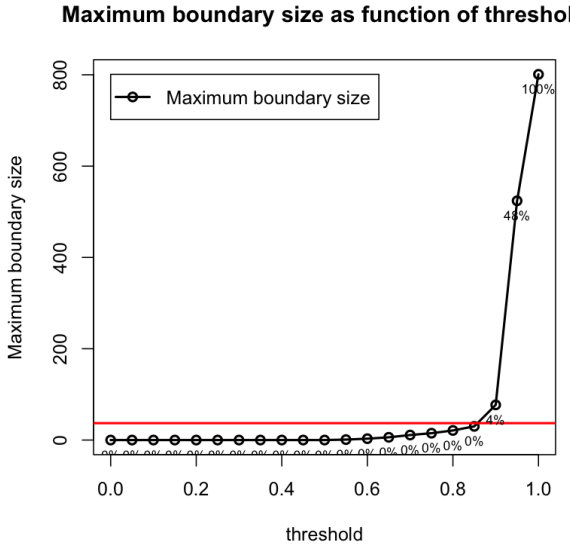


Figure 7: Maximum boundary as function of non-rejection rate cut-offs. The red line indicates the sample size of the leukemia data set (n =37) and next to each point the graph density is indicated.

This function also returns the largest cut-off β^*, among those considered in the plot, whose resulting estimated graph $\hat{G}$ has $bd(\hat{G}) < n$. In this case β^* =0.85 and using this cut-off we obtain a resulting qp-graph $\hat{G}^{(q)}$ which has $bd(\hat{G}^{(q)})$ =30 as shown here below:

```
> g <- qpGraph(gennrr, threshold=qpbd$threshold, return.type="graphNEL")
> g

A graphNEL graph with undirected edges
Number of Nodes = 707
Number of Edges = 1323
```

```
> max(degree(g))
```

```
[1] 30
```

Since $bd(\hat{G}^{(q)})$ =30 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:

$$\rho_{ij.R} = \frac{-\kappa_{ij}}{\sqrt{\kappa_{ii}\,\kappa_{jj}}} \text{ where } R = V \backslash \{i, j\}\,. \tag{7}$$

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()`:

```
> pac <- qpPAC(leukemia_canonicalPwy_es, g, return.K=TRUE, tol=0.01,
+              verbose=FALSE)
```

We employ the estimated PACs and their p-values to select a final estimate $\hat{G}$ of the underlying undirected graph G whose FDR of wrongly included edges is below a desired network-wide significance level. This FDR control at network-wide level helps in discarding spurious associations with a large marginal strength (i.e., a large Pearson correlation coefficient) but which in fact are indirectly occurring. Note below that PCCs are not directly estimated from the data but by scaling the MLE of the sample covariance matrix, obtaining in this way more precise estimates of the PCCs.

```
> gAM <- qpGraph(gennrr, threshold=qpbd$threshold)
> ridx <- row(pac$P)[as.matrix(upper.tri(pac$P) & gAM)]
> cidx <- col(pac$P)[as.matrix(upper.tri(pac$P) & gAM)]
> allEdges <- data.frame(PWYi=colnames(pac$P)[ridx],
+                        PWYj=colnames(pac$P)[cidx],
+                        PAC=pac$R[cbind(ridx, cidx)],
+                        PAC.P.value=pac$P[cbind(ridx, cidx)],
+                        PAC.adj.P.value=p.adjust(pac$P[cbind(ridx, cidx)], method="fdr"),
+                        PCC=cov2cor(solve(pac$K))[cbind(ridx, cidx)])
> allEdges <- allEdges[sort(allEdges$PAC.adj.P.value, index.return=TRUE)$ix, ]
> dim(allEdges)
```

```
[1] 1323    6
```

From the 1323 associations we select those with an absolute PCC larger than 0.7 and a PAC p-value leading to a network-wide FDR below 10%:

```
> sigEdges <- allEdges[which(allEdges$PAC.adj.P.value < 0.1 & abs(allEdges$PCC) > 0.7) , ]
> dim(sigEdges)
```

```
[1] 234   6
```

Using these significant edges we build a *graphNEL* object representing this network:

```
> vtc <- unique(as.character(unlist(sigEdges[, c("PWYi", "PWYj")], use.names=FALSE)))
> g <- new("graphNEL", nodes=vtc, edgemode="undirected")
> g <- addEdge(from=as.character(sigEdges[, "PWYi"]),
+              to=as.character(sigEdges[, "PWYj"]),
+              graph=g)
> g
```

```
A graphNEL graph with undirected edges
Number of Nodes = 255
Number of Edges = 234
```

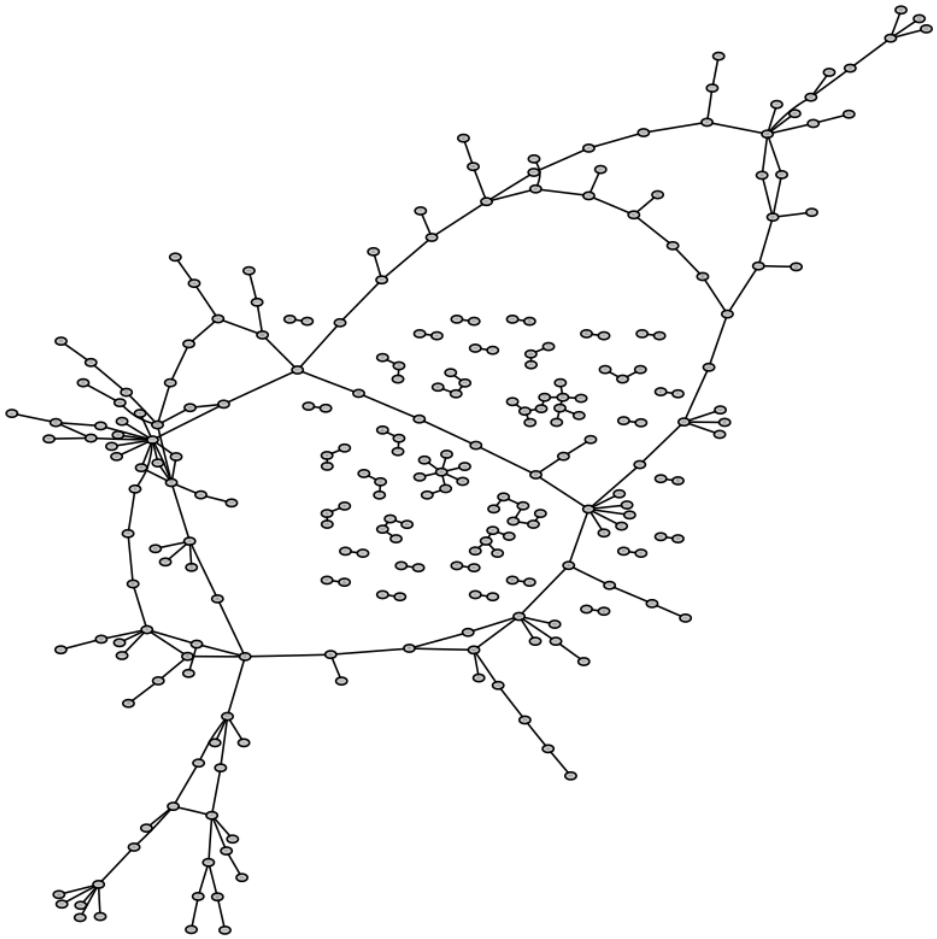


Figure 8: Network of cross-talk associations between Broad C2 canonical pathways of the leukemia data set obtained by a Gaussian graphical modeling approach by which edges are included in the graph with a minimum absolute PCC > 0.7 at a FDR $< 10\%$.

The network is divided in connected components of the following sizes:

```
> cct <- table(listLen(connComp(g)))
> cct

  2   3   4   5   6   8  11 154
 21   7   2   1   1   1   1   1
```

Thus, there is one very large connected component of 154 pathways in which we are going to focus our analysis. In order to dissect the larger component of 154 pathways, we are going to look to its degree distribution:

```
> vtcCCmax <- connComp(g)[[which(sapply(connComp(g), length) == max(as.integer(names(cct))))]]
> gCCmax <- subGraph(vtcCCmax, g)
> gCCmax

A graphNEL graph with undirected edges
Number of Nodes = 154
Number of Edges = 167

> table(degree(gCCmax))
```

```
 1  2  3  4  5  6  7  8 13
64 50 21  6  6  3  2  1  1
```

We can observe that four pathways, representing less than 3% of the total, have a connectivity degree higher than 6. These highly connected pathways are the following:

```
> sort(degree(gCCmax), decreasing=TRUE)[1:4]

                            BIOCARTA_MCM_PATHWAY
                                              13
                     REACTOME_NCAM1_INTERACTIONS
                                               8
                   KEGG_ONE_CARBON_POOL_BY_FOLATE
                                               7
REACTOME_NUCLEAR_RECEPTOR_TRANSCRIPTION_PATHWAY
                                               7
```

We are going to display the subnetwork formed by these highly-connected pathways, the paths that connect them and their interacting partners. We start by extracting the corresponding subgraph:

```
> vtcHubNet <- names(sort(degree(gCCmax), decreasing=TRUE)[1:4])
> pairs <- combn(vtcHubNet, 2)
> sp <- sp.between(gCCmax, pairs[1, ], pairs[2, ])
> vtcInShortestPaths <- unlist(sapply(sp, function(x) x$path_detail), use.names=FALSE)
> vtcHubNet <- unique(c(vtcHubNet,
+                       unlist(boundary(vtcHubNet, gCCmax), use.names=FALSE),
+                       vtcInShortestPaths))
> gHubNet <- subGraph(vtcHubNet, gCCmax)
> gHubNet

A graphNEL graph with undirected edges
Number of Nodes = 50
Number of Edges = 54
```

The resulting network is shown in Figure 9. The gene sets connected by the meta-pathway analysis show a clear structure, forming four distinct groups with each having a hub gene set. The gene sets surrounding each hub gene set are functionally related to each other and associated with cancer. Most of the gene sets are related to genome stability, cell cycle and cell proliferation, have an effect on the regulation of cell differentiation processes, and pertain to the escape of apoptosis. Also, the members of the *NUCLEAR RECEPTOR TRANSCRIPTION PATHWAY* -estrogen receptors, retinoic acid receptor and nuclear receptors- have been indicated as potential drug targets in disease Gronemeyer et al. (2004). Among them, Nr4a3 and Nr4a1 are proteins whose function is not very well understood. However, they have been implicated as tumor suppressors in myeloid leukemogenesis Mullican et al. (2007). Also, the decreased expression of retinoic acid receptor α plays a role in leukemogenesis Glasow et al. (2008). In fact, a recent review enumerating all genes known to be involved in MLL, includes nuclear-, retinoic acid- and estrogen receptors Ansari and Mandal (2010). The *FGFR LIGAND BINDING AND ACTIVATION* pathway contains members of the fibroblast growth factor family which are involved in oncogenic mechanisms Turner and Grose (2010) and have been investigated as therapeutic targets Greulich and Pollock (2011).

```
> nodlab <- gsub("_", " ", gsub("KEGG_|REACTOME_|BIOCARTA_", "", vtcHubNet))
> nodlab <- sapply(nodlab, function(x) { v <- unlist(strsplit(x, ' ')) ; t <- ""; l <- 0; for (w in
> names(nodlab) <- vtcHubNet
> nodeRenderInfo(gHubNet) <- list(shape="ellipse", label=nodlab, fill="lightgrey", lwd=1)
> edgeRenderInfo(gHubNet) <- list(lwd=1)
> gHubNet <- layoutGraph(gHubNet, layoutType="fdp")
> renderGraph(gHubNet)
```

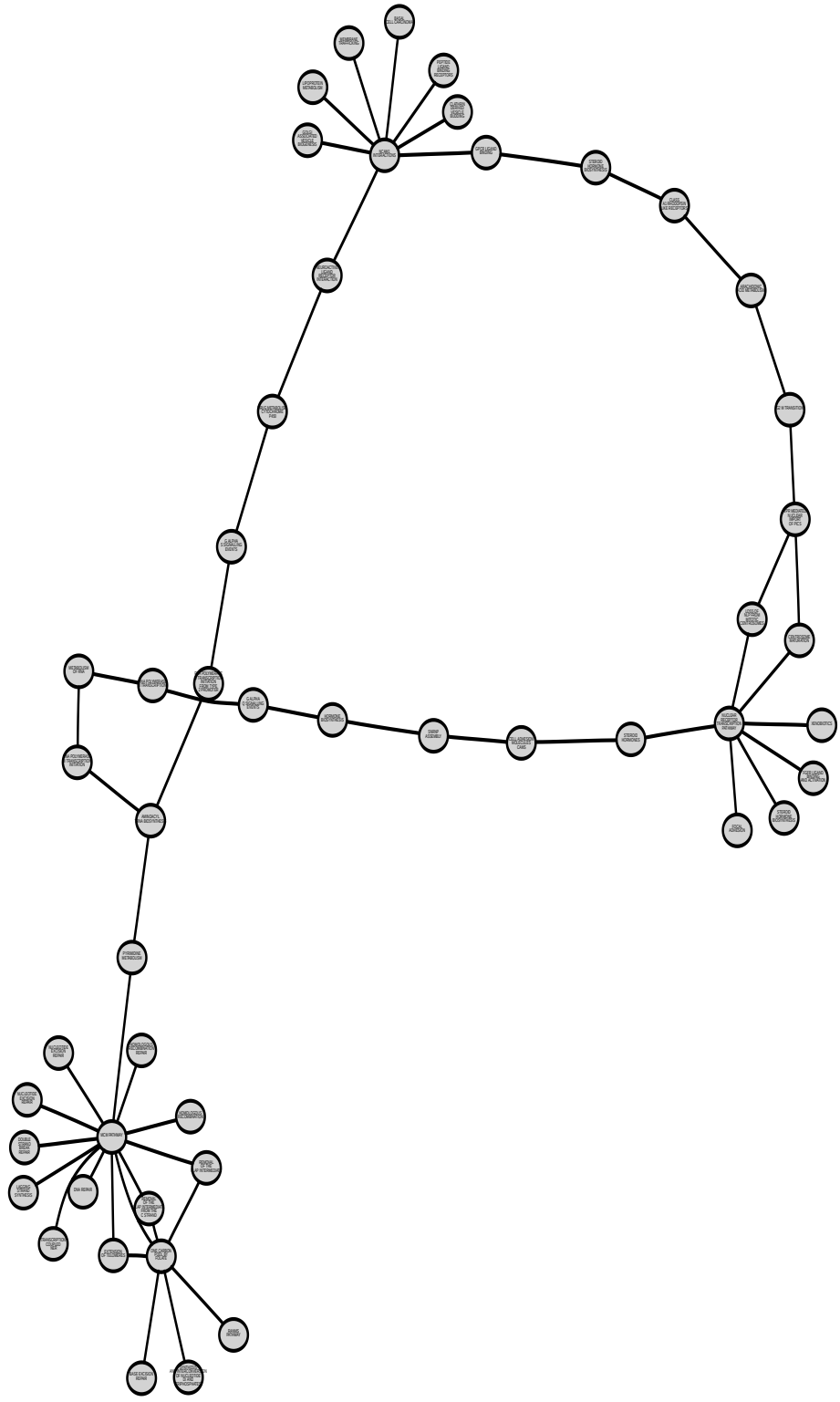


Figure 9: Network of cross-talk associations between Broad C2 canonical pathways of the leukemia data set obtained by a Gaussian graphical modeling approach by which edges are included in the graph with a minimum absolute PCC > 0.7 at a FDR $< 10\%$. The displayed network only includes edges connecting the 4 hub genes with highest degree and their interaction partners.

Another gene set that comes to attention is *ONE CARBON POOL BY FOLATE* because it is not directly associated with proliferation or mitosis. Genetic variation in folate metabolism has been reported to be associated with 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 the genes forming this KEGG pathway, the *MTR* gene encodes an enzyme that catalyzes the final step in methionine biosynthesis and has been associated to an increased risk of ALL. This risk was most pronounced for cases with the *MLL* translocation Lightfoot et al. (2010).

5 Comparison with other methods

In this section we compare with simulated data the performance of GSVA with other methods producing pathway summaries of gene expression, concretely, PLAGE, the combined z-score and ssGSEA which are available through the argument `method` of the function `gsva()`. We employ the following simple linear additive model for simulating normalized microarray data on p genes and n samples divided in two groups representing a case-control scenario:

$$y_{ij} = \alpha_i + \beta_j + \epsilon_{ij}\,, \tag{8}$$

where $\alpha_i \sim \mathcal{N}(0,1)$ is a gene-specific effect, such as a probe-effect, with $i = 1,\ldots,p$, $\beta_j \sim \mathcal{N}(\mu_j, 0.5)$ is a sample-effect with $j = 1, 2$ and $e_{ij} \sim \mathcal{N}(0, 0.1)$ corresponds to random noise.

We will assess the statistical power to detect one differentially expressed gene set formed by 30 genes, out of $p = 1000$, as a function of the sample size and two varying conditions: the fraction of differentially expressed genes in the gene set (50% and 80%) and the signal-to-noise ratio expressed as the magnitude of the mean sample effect for one of the sample groups ($\mu_1 = 0$ and either $\mu_2 = 0.2$ or $\mu_2 = 1$).

The following function enables simulating such data, computes the corresponding GSE scores with each method, performs a t-test on the tested gene set between the two groups of samples for each method and returns the corresponding p-values:

```
> runSim <- function(p, n, gs.sz, S2N, fracDEgs) {
+   sizeDEgs <- round(fracDEgs * gs.sz)
+   group.n <- round(n / 2)
+
+   sampleEffect <- rnorm(n, mean=0, sd=1)
+   sampleEffectDE <- rnorm(n, mean=S2N, sd=0.5)
+   probeEffect <- rnorm(p, mean=0, sd=1)
+   noise <- matrix(rnorm(p*n, mean=0, sd=0.1), nrow=p, ncol=n)
+   noiseDE <- matrix(rnorm(p*n, mean=0, sd=0.1), nrow=p, ncol=n)
+   M <- outer(probeEffect, sampleEffect, "+") + noise
+   M2 <- outer(probeEffect, sampleEffectDE, "+") + noiseDE
+   M[1:sizeDEgs, 1:group.n] <- M2[1:sizeDEgs, 1:group.n]
+
+   rownames(M) <- paste("g", 1:nrow(M), sep="")
+   geneSets <- list(testGeneSet=paste0("g", 1:(gs.sz)))
+
+   es.gsva <- gsva(M, geneSets, verbose=FALSE)$es.obs
+   es.plage <- gsva(M, geneSets, method="plage", verbose=FALSE)
+   es.zscore <- gsva(M, geneSets, method="zscore", verbose=FALSE)
```

```
+   es.ssgsea <- gsva(M, geneSets, method="ssgsea", verbose=FALSE)
+
+   gsva.pval <- t.test(es.gsva[1:group.n],es.gsva[(group.n+1):length(es.gsva)])$p.value
+   plage.pval <- t.test(es.plage[1:group.n],es.plage[(group.n+1):length(es.plage)])$p.value
+   zscore.pval <- t.test(es.zscore[1:group.n],es.zscore[(group.n+1):length(es.zscore)])$p.value
+   ssgsea.pval <- t.test(es.ssgsea[1:group.n],es.ssgsea[(group.n+1):length(es.ssgsea)])$p.value
+
+   c(gsva.pval, ssgsea.pval, zscore.pval, plage.pval)
+ }
```

The next function takes the p-values of the output of the previous function and estimates the statistical power as the fraction of non-rejections at a significant level $\alpha = 0.05$. It assumes the p-values were generated under the alternative hypothesis of differential expression.

```
> estimatePower <- function(pvals, alpha=0.05) {
+   N <- ncol(pvals)
+   c(1 - sum(pvals[1, ] > alpha)/N, 1 - sum(pvals[2, ] > alpha)/N,
+     1 - sum(pvals[3, ] > alpha)/N, 1 - sum(pvals[4, ] > alpha)/N)
+ }
```

Finally, we perform the simulation on each of the four described scenarios 50 times using the code below. The results in Fig. 10 show that GSVA attains higher statistical power than the other three methods in each of the simulated scenarios.

```
> set.seed(1234)
> exp1 <- cbind(estimatePower(replicate(50, runSim(1000, 10, gs.sz=30, S2N=0.2, fracDEgs=0.5))),
+               estimatePower(replicate(50, runSim(1000, 20, gs.sz=30, S2N=0.2, fracDEgs=0.5))),
+               estimatePower(replicate(50, runSim(1000, 40, gs.sz=30, S2N=0.2, fracDEgs=0.5))),
+               estimatePower(replicate(50, runSim(1000, 60, gs.sz=30, S2N=0.2, fracDEgs=0.5))))
> exp2 <- cbind(estimatePower(replicate(50, runSim(1000, 10, gs.sz=30, S2N=1.0, fracDEgs=0.5))),
+               estimatePower(replicate(50, runSim(1000, 20, gs.sz=30, S2N=1.0, fracDEgs=0.5))),
+               estimatePower(replicate(50, runSim(1000, 40, gs.sz=30, S2N=1.0, fracDEgs=0.5))),
+               estimatePower(replicate(50, runSim(1000, 60, gs.sz=30, S2N=1.0, fracDEgs=0.5))))
> exp3 <- cbind(estimatePower(replicate(50, runSim(1000, 10, gs.sz=30, S2N=0.2, fracDEgs=0.8))),
+               estimatePower(replicate(50, runSim(1000, 20, gs.sz=30, S2N=0.2, fracDEgs=0.8))),
+               estimatePower(replicate(50, runSim(1000, 40, gs.sz=30, S2N=0.2, fracDEgs=0.8))),
+               estimatePower(replicate(50, runSim(1000, 60, gs.sz=30, S2N=0.2, fracDEgs=0.8))))
> exp4 <- cbind(estimatePower(replicate(50, runSim(1000, 10, gs.sz=30, S2N=1.0, fracDEgs=0.8))),
+               estimatePower(replicate(50, runSim(1000, 20, gs.sz=30, S2N=1.0, fracDEgs=0.8))),
+               estimatePower(replicate(50, runSim(1000, 40, gs.sz=30, S2N=1.0, fracDEgs=0.8))),
+               estimatePower(replicate(50, runSim(1000, 60, gs.sz=30, S2N=1.0, fracDEgs=0.8))))
```

6 GSVA for RNA-Seq data

In this section we illustrate how to use GSVA with RNA-seq data and, more importantly, how the method provides pathway activity profiles analogous to the ones obtained from microarray data by using samples of lymphoblastoid cell lines (LCL) from HapMap individuals which have been profiled using both technologies Huang et al. (2007); Pickrell et al. (2010). These data form part of the experimental package GSVAdata and the corresponding help pages contain details on how the data were processed. We start loading these data and verifying that they indeed contain expression data for the same genes and samples, as follows:

```
> data(commonPickrellHuang)
> stopifnot(identical(featureNames(huangArrayRMAnoBatchCommon_eset),
+                     featureNames(pickrellCountsArgonneCQNcommon_eset)))
> stopifnot(identical(sampleNames(huangArrayRMAnoBatchCommon_eset),
+                     sampleNames(pickrellCountsArgonneCQNcommon_eset)))
```

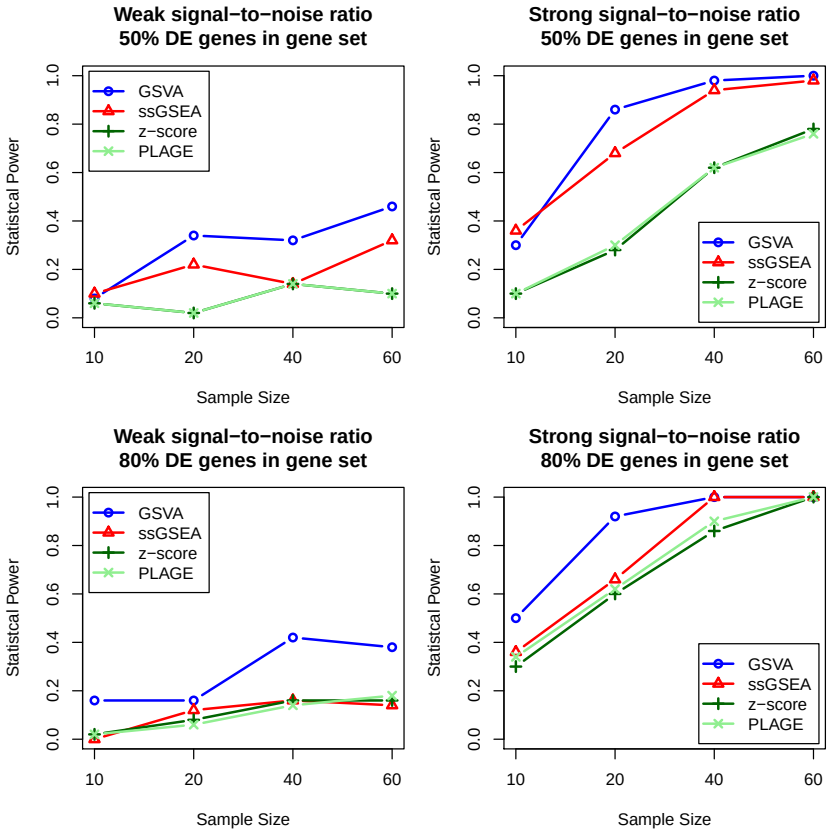


Figure 10: **Comparison of the statistical power of GSVA, PLAGE, single sample GSEA (ssGSEA) and combined z-score (zscore).** Each panel shows the statistical power in the y-axis as function of the sample size in the x-axis. GSE scores were calculated with each method with respect to a single DE gene set. The two panels on top correspond to simulations where 50% of the genes in the gene set were DE while the two at the bottom contained 80% of DE genes. The two panels on the left correspond to a weak signal-to-noise ratio in the DE magnitude while the two on the right correspond to a strong one. Data were simulated from a linear model with additive sample and probe effects reflecting the previous four different scenarios. The statistical power was estimated as 1 minus the fraction of non-rejections on the DE gene set at $\alpha = 0.05$. GSVA provides in general higher power than the other three methods, specially under a weak signal-to-noise ratio.

Next, for the current analysis we use the previously defined collection of canonical C2 Broad Sets extended with two gene sets formed by genes with sex-specific expression:

```
> data(genderGenesEntrez)
> MSY <- GeneSet(msYgenesEntrez, geneIdType=EntrezIdentifier(),
+                collectionType=BroadCollection(category="c2"), setName="MSY")
> MSY

setName: MSY
geneIds: 266, 84663, ..., 353513 (total: 34)
geneIdType: EntrezId
collectionType: Broad
  bcCategory: c2 (Curated)
  bcSubCategory:  NA
details: use 'details(object)'

> XiE <- GeneSet(XiEgenesEntrez, geneIdType=EntrezIdentifier(),
+                collectionType=BroadCollection(category="c2"), setName="XiE")
```

```
> XiE

setName: XiE
geneIds: 293, 8623, ..., 1121 (total: 66)
geneIdType: EntrezId
collectionType: Broad
  bcCategory: c2 (Curated)
  bcSubCategory:  NA
details: use 'details(object)'

> canonicalC2BroadSets <- GeneSetCollection(c(canonicalC2BroadSets, MSY, XiE))
> canonicalC2BroadSets

GeneSetCollection
  names: KEGG_GLYCOLYSIS_GLUCONEOGENESIS, KEGG_CITRATE_CYCLE_TCA_CYCLE, ..., XiE (835 total)
  unique identifiers: 55902, 2645, ..., 1121 (6810 total)
  types in collection:
    geneIdType: EntrezIdentifier (1 total)
    collectionType: BroadCollection (1 total)
```

We calculate now GSVA enrichment scores for these gene sets using first the microarray data and then the RNA-seq data. Note that the only requirement to do the latter is to set the argument `rnaseq=TRUE` which is `FALSE` by default.

```
> esmicro <- gsva(huangArrayRMAnoBatchCommon_eset, canonicalC2BroadSets, min.sz=5, max.sz=500,
+                 mx.diff=TRUE, verbose=FALSE, rnaseq=FALSE)$es.obs
> dim(esmicro)

Features  Samples
     806       36

> esrnaseq <- gsva(pickrellCountsArgonneCQNcommon_eset, canonicalC2BroadSets, min.sz=5, max.sz=500,
+                  mx.diff=TRUE, verbose=FALSE, rnaseq=TRUE)$es.obs
> dim(esrnaseq)

Features  Samples
     806       36
```

To compare expression values from both technologies we are going to transform the RNA-seq read counts into RPKM values. For this purpose we need gene length and G+C content information also stored in the GSVAdata package and use the `cpm()` function from the edgeR package. Note that RPKMs can only be calculated for those genes for which the gene length and G+C content information is available:

```
> data(annotEntrez220212)
> head(annotEntrez220212)

          Length GCcontent
1           2301 0.6292916
10          1344 0.3816964
100         2612 0.5153139
1000        4380 0.4502283
10000       7091 0.3989564
100008586    606 0.4339934

> cpm <- edgeR::cpm(exprs(pickrellCountsArgonneCQNcommon_eset))
> dim(cpm)

[1] 11508    36
```

```
> common <- intersect(rownames(cpm), rownames(annotEntrez220212))
> length(common)

[1] 11478

> rpkm <- sweep(cpm[common, ], 1, annotEntrez220212[common, "Length"] / 10^3, FUN="/")
> dim(rpkm)

[1] 11478    36

> dim(huangArrayRMAnoBatchCommon_eset[rownames(rpkm), ])

Features  Samples
   11478       36
```

We finally calculate Spearman correlations between gene and gene-level expression values and gene set level GSVA enrichment scores produced from data obtained by microarray and RNA-seq technologies:

```
> corsrowsgene <- sapply(1:nrow(huangArrayRMAnoBatchCommon_eset[rownames(rpkm), ]),
+                        function(i, expmicro, exprnaseq) cor(expmicro[i, ], exprnaseq[i, ], method=
+                        exprs(huangArrayRMAnoBatchCommon_eset[rownames(rpkm), ]), log2(rpkm+0.1))
> names(corsrowsgene) <- rownames(rpkm)
> corsrowsgs <- sapply(1:nrow(esmicro),
+                      function(i, esmicro, esrnaseq) cor(esmicro[i, ], esrnaseq[i, ], method="spear
+                      exprs(esmicro), exprs(esrnaseq))
> names(corsrowsgs) <- rownames(esmicro)
```

In panels A and B of Figure 11 we can see the distribution of these correlations at gene and gene set level. They show that GSVA enrichment scores correlate similarly to gene expression levels produced by both profiling technologies.

We also examined the two gene sets containing gender specific genes in detail: those that escape X-inactivation in female samples (Carrel and Willard, 2005) and those that are located on the male-specific region of the Y chrosomome (Skaletsky et al., 2003). In panels C and D of Figure 11 we can see how microarray and RNA-seq enrichment scores correlate very well in these gene sets, with $\rho = 0.82$ for the male-specific gene set and $\rho = 0.78$ for the female-specific gene set. Male and female samples show higher GSVA enrichment scores in their corresponding gene set. This demonstrates the flexibility of GSVA to enable analogous unsupervised and single sample GSE analyses in data coming from both, microarray and RNA-seq technologies.

7 Session Information

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

- R version 2.15.1 (2012-06-22), `i386-apple-darwin9.8.0`
- Locale: `C`
- Base packages: base, datasets, grDevices, graphics, grid, methods, stats, utils
- Other packages: AnnotationDbi 1.18.3, Biobase 2.16.0, BiocGenerics 0.2.0, DBI 0.2-5, GSEABase 1.18.0, GSVA 1.4.4, GSVAdata 0.99.8, RBGL 1.32.1, RColorBrewer 1.0-5, RSQLite 0.11.2, Rgraphviz 1.34.2, annotate 1.34.1, genefilter 1.38.0, graph 1.34.0, hgu95a.db 2.7.1, limma 3.12.3, org.Hs.eg.db 2.7.1, qpgraph 1.12.2
- Loaded via a namespace (and not attached): GGBase 3.18.0, IRanges 1.14.4, Matrix 1.0-9, XML 3.9-4, edgeR 2.6.12, lattice 0.20-10, snpStats 1.6.0, splines 2.15.1, stats4 2.15.1, survival 2.36-14, tools 2.15.1, xtable 1.7-0

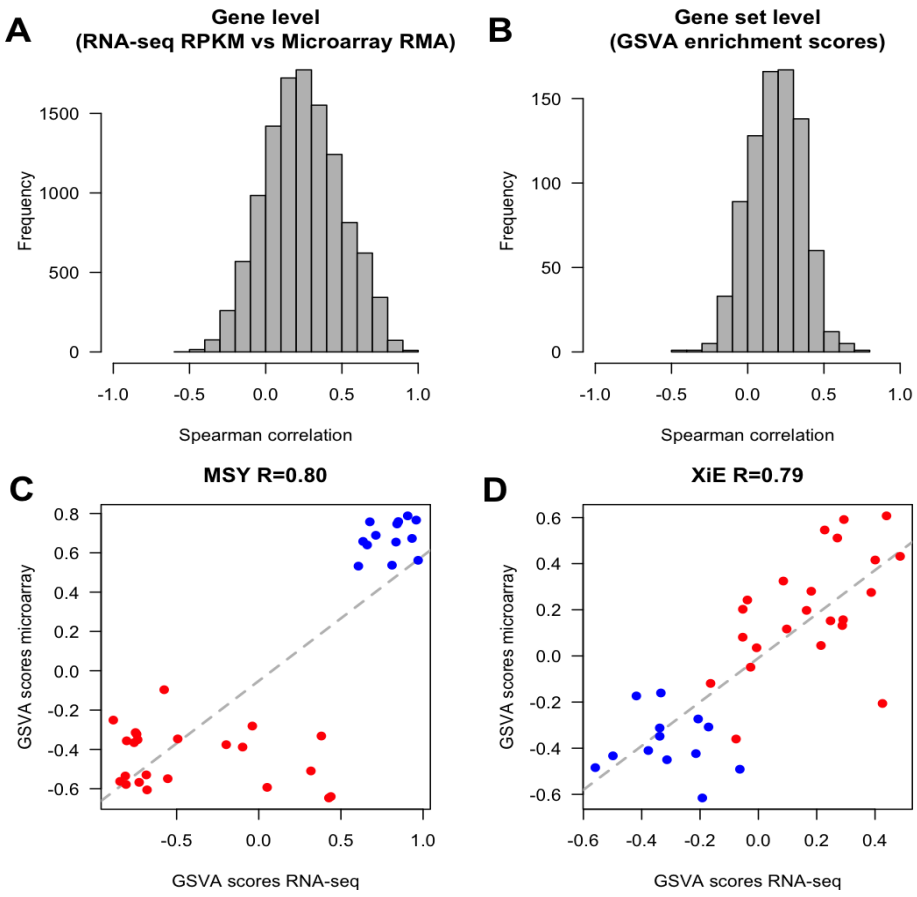


Figure 11: **GSVA for RNA-seq (Argonne).** A. Distribution of Spearman correlation values between gene expression profiles of RNA-seq and microarray data. B. Distribution of Spearman correlation values between GSVA enrichment scores of gene sets calculated from RNA-seq and microarray data. C and D. Comparison of GSVA enrichment scores obtained from microarray and RNA-seq data for two gene sets containing genes with sex-specific expression: MSY formed by genes from the male-specific region of the Y chromosome, thus male-specific, and XiE formed by genes that escape X-inactivation in females, thus female-specific. Red and blue dots represent female and male samples, respectively. In both cases GSVA-scores show very high correlation between the two profiling technologies where female samples show higher enrichment scores in the female-specific gene set and male samples show higher enrichment scores in the male-specific gene set.

References

Ansari, K. I. and Mandal, S. S. (2010). Mixed lineage leukemia: roles in gene expression, hormone signaling and mRNA processing. *FEBS Journal*, 277(8):1790–1804.

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.

Barbie, D. A., Tamayo, P., Boehm, J. S., Kim, S. Y., Moody, S. E., Dunn, I. F., Schinzel, A. C., Sandy, P., Meylan, E., Scholl, C., Fr\textbackslashtextbar[ouml]\textbackslashtextbarhling, S., Chan, E. M., Sos, M. L., Michel, K., Mermel, C., Silver, S. J., Weir, B. A., Reiling, J. H., Sheng, Q., Gupta, P. B., Wadlow, R. C., Le, H., Hoersch, S., Wittner, B. S., Ramaswamy, S., Livingston, D. M., Sabatini, D. M., Meyerson, M., Thomas, R. K., Lander, E. S., Mesirov, J. P., Root, D. E., Gilliland, D. G., Jacks, T., and Hahn, W. C. (2009). Systematic RNA interference reveals that oncogenic KRAS-driven cancers require TBK1. *Nature*, 462(7269):108–112.

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.

Canale, A. and Dunson, D. B. (2011). Bayesian kernel mixtures for counts. *Journal of the American Statistical Association*, 106(496):1528–1539.

Carrel, L. and Willard, H. F. (2005). X-inactivation profile reveals extensive variability in x-linked gene expression in females. *Nature*, 434(7031):400–404.

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.

Edelman, E., Porrello, A., Guinney, J., Balakumaran, B., Bild, A., Febbo, P. G., and Mukherjee, S. (2006). Analysis of sample set enrichment scores: assaying the enrichment of sets of genes for individual samples in genome-wide expression profiles. *Bioinformatics*, 22(14):e108–e116.

Glasow, A., Barrett, A., Petrie, K., Gupta, R., Boix-Chornet, M., Zhou, D., Grimwade, D., Gallagher, R., von Lindern, M., Waxman, S., Enver, T., Hildebrandt, G., and Zelent, A. (2008). DNA methylation-independent loss of RARA gene expression in acute myeloid leukemia. *Blood*, 111(4):2374 –2377.

Greulich, H. and Pollock, P. M. (2011). Targeting mutant fibroblast growth factor receptors in cancer. *Trends in Molecular Medicine*, 17(5):283–292.

Gronemeyer, H., Gustafsson, J., and Laudet, V. (2004). Principles for modulation of the nuclear receptor superfamily. *Nat Rev Drug Discov*, 3(11):950–964.

Hansen, K. D., Irizarry, R. A., and Wu, Z. (2012). Removing technical variability in RNA-seq data using conditional quantile normalization. *Biostatistics*.

Hänzelmann, S., Castelo, R., and Guinney, J. (2012). GSVA: Gene set variation analysis for microarray and rna-seq data. *submitted*.

Huang, R. S., Duan, S., Bleibel, W. K., Kistner, E. O., Zhang, W., Clark, T. A., Chen, T. X., Schweitzer, A. C., Blume, J. E., Cox, N. J., and Dolan, M. E. (2007). A genome-wide approach to identify genetic variants that contribute to etoposide-induced cytotoxicity. *Proceedings of the National Academy of Sciences of the United States of America*, 104(23):9758–9763.

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

Lee, E., Chuang, H., Kim, J., Ideker, T., and Lee, D. (2008). Inferring pathway activity toward precise disease classification. *PLoS Comput Biol*, 4(11):e1000217.

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 Study, U. K. C. C. (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.

Mullican, S. E., Zhang, S., Konopleva, M., Ruvolo, V., Andreeff, M., Milbrandt, J., and Conneely, O. M. (2007). Abrogation of nuclear receptors nr4a3 andNr4a1 leads to development of acute myeloid leukemia. *Nat Med*, 13(6):730–735.

Pearson, E. (1963). Comparison of tests for randomness of points on a line. *Biometrika*, 50:315–325.

Pickrell, J. K., Marioni, J. C., Pai, A. A., Degner, J. F., Engelhardt, B. E., Nkadori, E., Veyrieras, J., Stephens, M., Gilad, Y., and Pritchard, J. K. (2010). Understanding mechanisms underlying human gene expression variation with RNA sequencing. *Nature*, 464(7289):768–772.

Roverato, A. and Castelo, R. (2010). Learning undirected graphical models from multiple datasets with the generalized non-rejection rate. In Myllymäki, P., Roos, T., and Jaakkola, T., editors, *Proceedings of the Fifth European Workshop on Probabilistic Graphical Models*, pages 249–56, Helsinki. HIIT Publications.

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.

Skaletsky, H., Kuroda-Kawaguchi, T., Minx, P. J., Cordum, H. S., Hillier, L., Brown, L. G., Repping, S., Pyntikova, T., Ali, J., Bieri, T., Chinwalla, A., Delehaunty, A., Delehaunty, K., Du, H., Fewell, G., Fulton, L., Fulton, R., Graves, T., Hou, S., Latrielle, P., Leonard, S., Mardis, E., Maupin, R., McPherson, J., Miner, T., Nash, W., Nguyen, C., Ozersky, P., Pepin, K., Rock, S., Rohlfing, T., Scott, K., Schultz, B., Strong, C., Tin-Wollam, A., Yang, S., Waterston, R. H., Wilson, R. K., Rozen, S., and Page, D. C. (2003). The male-specific region of the human y chromosome is a mosaic of discrete sequence classes. *Nature*, 423(6942):825–837.

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.

Tomfohr, J., Lu, J., and Kepler, T. B. (2005). Pathway level analysis of gene expression using singular value decomposition. *BMC Bioinformatics*, 6:225. PMID: 16156896 PMCID: PMC1261155.

Turner, N. and Grose, R. (2010). Fibroblast growth factor signalling: from development to cancer. *Nat Rev Cancer*, 10(2):116–129.

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.

Zilliox, M. J. and Irizarry, R. A. (2007). A gene expression bar code for microarray data. *Nat Meth*, 4(11):911–913.