

Tutorial:  
“*gCMAP*: user-friendly connectivity mapping with R”

Thomas Sandmann, Sarah Kummerfeld, Robert Gentleman and Richard Bourgon

July 28, 2013

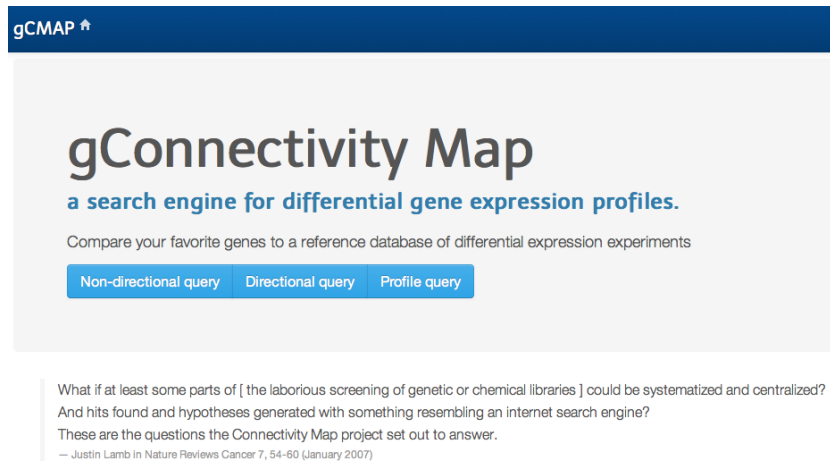
## Contents

<b>1</b>	<b>Quickstart</b>	<b>2</b>
1.1	Submitting queries . . . . .	2
1.1.1	Querying with genes sets . . . . .	2
1.1.2	Submitting differential expression profiles . . . . .	4
1.1.3	Performing queries in the command line . . . . .	4
<b>2</b>	<b>A small connectivity map for human HepG2 cells</b>	<b>5</b>
2.1	Retrieve raw microarray data from ArrayExpress . . . . .	5
2.2	Create a connectivity map . . . . .	5
2.3	Obtaining public gene set annotations . . . . .	7
2.4	Starting the <i>gCMAPWeb</i> application . . . . .	7
<b>3</b>	<b>RNAseq analysis of the benzo[a]pyrene response in HepG2 cells</b>	<b>8</b>
3.1	Differential expression analysis of RNAseq data . . . . .	8
3.2	Connectivity mapping with <i>gCMAPWeb</i> . . . . .	10
3.2.1	Performing a non-directional query . . . . .	12
3.2.2	Connectivity mapping in the command line . . . . .	13
<b>4</b>	<b>Building and querying the Broad connectivity Map</b>	<b>16</b>
4.1	Differential expression analysis . . . . .	16
4.2	Querying the Broad connectivity map with <i>gCMAPWeb</i> . . . . .	17
4.2.1	Non-directional query: assessing overlap between query and reference datasets . . . . .	18
4.2.2	Directional query: using information about the direction of gene expression changes . . . . .	18
4.2.3	Profile query: submitting differential expression scores . . . . .	20
4.3	Queries in the command line . . . . .	24
4.3.1	Non-directional queries . . . . .	24
4.3.2	Directional queries . . . . .	24
<b>5</b>	<b>Quality control</b>	<b>25</b>

# 1 Quickstart

To start a local instance of the *gCMAPWeb* application on your system, start R, install the *gCMAP* and *gCMAPWeb* packages and type

```
> library( gCMAPWeb )  
> gCMAPWeb()
```



**Figure 1:** Screenshot of the *gCMAPWeb* index page, featuring links to perform non-directional, directional and profile queries.

A *gCMAPWeb* application instance populated with small, simulated datasets will open in your default browser, using R’s internal web server. To evaluate *gCMAPWeb*’s functionality, choose one of the four query types, *Gene lookup*, *Non-directional query*, *Directional query* or *Profile query* to proceed to the respective query submission page. (See page 11 for more details on the different query types.)

## 1.1 Submitting queries

On each submission page, the “Example query” button below the text field will populate the text field with a suitable query for the simulated datasets.

### 1.1.1 Querying with genes sets

For example, to identify experimental conditions affecting a directional gene set of interest, e.g. genes observed to be up- or down-regulated in a previous study, choose the *Directional query* option (see page 11). *gCMAPWeb*’s query submission page is shown in (Figure 2). Clicking the *Example query* button prefills the form with two sets of Entrez gene identifiers (matching the simulated example datasets).

Alternatively, you can also specify genes by providing HUGO gene symbols or microarray probe identifiers. Ticking the *Gene Symbol* or *Probe identifier* radio button, respectively, will prompt *gCMAPWeb* to automatically retrieve the corresponding Entrez identifiers for you (Figure 3).

For convenience, longer gene lists can be uploaded as text files, containing identifiers separated by tab stops, commata or semicolons.

Next, choose one or more reference datasets to query and press *Submit*.

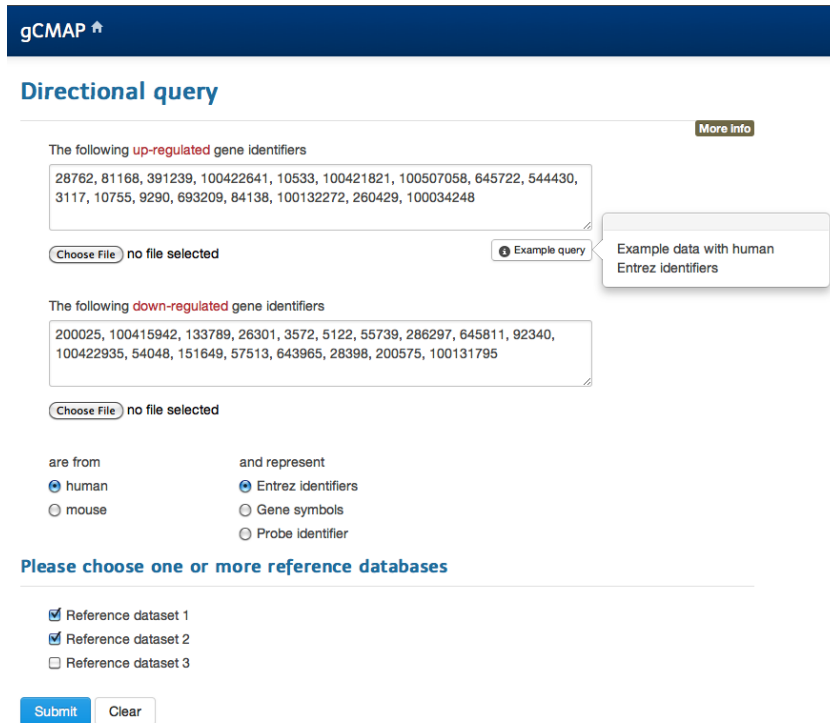


Figure 2: Screenshot of the *gCMAPWeb* query submission page for directional queries.

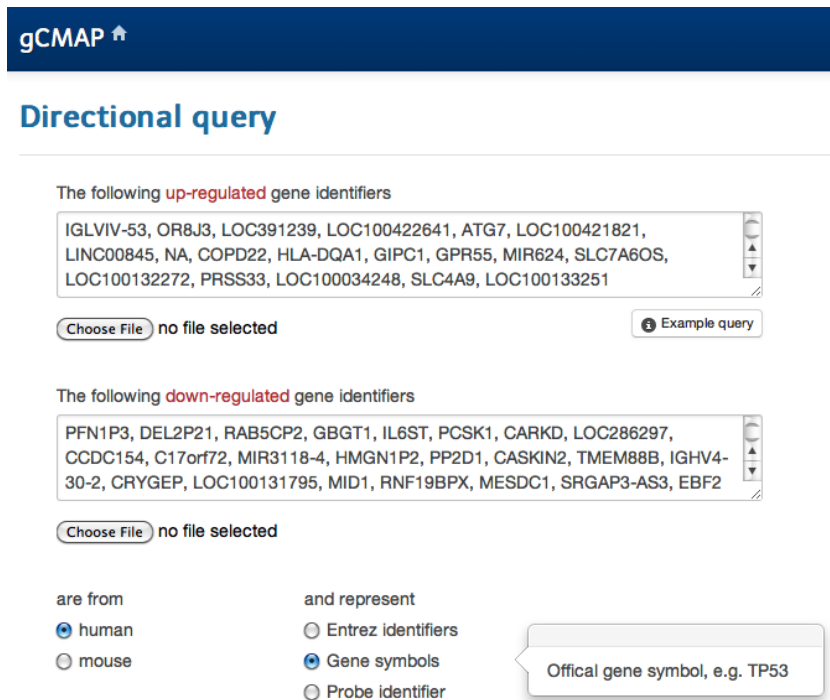


Figure 3: Detail of the *gCMAPWeb* submission page: Entering HUGO gene symbols

### 1.1.2 Submitting differential expression profiles

To perform *Profile queries* (see page 11), users can provide their own quantitative differential expression data. To see an example, choose the *Profile query* option on the *gCMAPWeb* start page.

Each gene identifier is accompanied by a single score in the same input line, e.g. a z-score indicating the significance of the observed differential gene expression (Figure 4). Alternatively, the same information can be uploaded in a text file.

The screenshot shows the 'Profile query' submission page on gCMAPWeb. It features a dark blue header with the 'gCMAP' logo. The main content area is titled 'Profile query' and includes a 'More info' link. A text input field contains the text 'The following gene identifiers and expression scores' followed by a list of three entries: '28762, 2.44', '81168, 2.77', and '391239, 4.56'. Below the text input is a 'Choose File' button and the text 'no file selected'. To the right of the text input is an 'Example query' button with a tooltip that says 'Example data with human Entrez identifiers and z-scores'. Below these are two columns of radio buttons: 'are from' with 'human' (selected) and 'mouse'; and 'and represent' with 'Entrez identifiers' (selected), 'Gene symbols', and 'Probe identifier'. Below this is the instruction 'Please choose one or more reference databases' followed by four checkboxes: 'Reference dataset 1' (checked), 'Reference dataset 2' (unchecked), 'Reference dataset 3' (checked), and 'Reference dataset 5' (unchecked). At the bottom are 'Submit' and 'Clear' buttons.

**Figure 4:** The *gCMAPWeb* Profile query submission page: Gene identifiers are accompanied by numeric scores, e.g. z-scores.

More information about the different fields of *gCMAPWeb*'s query submission page is available via the *Help* link in the top right corner of each web page.

### 1.1.3 Performing queries in the command line

All queries can also be performed without the graphical user interface. The *gCMAP* provides methods to use (or coerce) well-established gene set objects, including `GeneSet` and `GeneSetCollection` objects from the *GSEABase* package and offers the `CMAPCollection` class for efficient storage of large gene set collections.

Please see page 13 for examples and refer to the *gCMAP* documentation for details about specific methods.

## 2 A small connectivity map for human HepG2 cells

Kawata et al. used microarrays to profile the transcriptional response of human HepG2 cells upon treatment with 2,3-Dimethoxy-1,4-naphthoquinone, N-nitrosodimethylamine, phenol and six heavy metals [4]. (PubMed accession 17547211, ArrayExpress accession E-GEOD-6907.)

### 2.1 Retrieve raw microarray data from ArrayExpress

```
> library( ArrayExpress )
> library( affy )
> library( gCMAP )
> library( hgfocus.db )
```

The authors deposited the raw data at ArrayExpress from where it can be retrieved with the `ArrayExpress` function from the *ArrayExpress* package. Ca. **25 Mb** of data will be downloaded.

```
> GEO6907.batch <- ArrayExpress( "E-GEOD-6907" )
```

The experiments were performed using Affymetrix microarrays. We normalize the intensities and summarize the probesets with the `rma` function from the *affy* package.

```
> GEO6907.eSet <- rma( GEO6907.batch )
```

### 2.2 Create a connectivity map

Some genes are represented by multiple probesets on the array. We map all probe identifiers to Entrez IDs and calculate the mean expression for each gene in one step.

```
> GEO6907.eSet <- mapNmerge( GEO6907.eSet )
```

The `phenoData` slot of the `GEO6907.eSet ExpressionSet` contains various annotations columns. Experimental factors can be identified by their `Factor` prefix.

```
> conditions <- grep( "^Factor", varLabels( GEO6907.eSet ), value=TRUE )
> conditions
```

```
[1] "Factor.Value..Dose."      "Factor.Value..COMPOUND."
```

```
> pData( GEO6907.eSet ) <- pData( GEO6907.eSet )[,conditions]
> head( pData( GEO6907.eSet ) )
```

	Factor.Value..Dose.	Factor.Value..COMPOUND.
GSM159338	10.0	2,3-dimethoxy-1,4-naphthoquinone
GSM159331	20.0	Mercury (II) chloride
GSM159318	2.0	Cadmium chloride 2.5-hydrate
GSM159325	6.5	Nickel (II) chloride hexahydrate
GSM159336	10.0	2,3-dimethoxy-1,4-naphthoquinone
GSM159332	20.0	Mercury (II) chloride

Now we can use the annotated experimental factors to split the dataset into individual perturbation instances. We are interested in the `COMPOUND` annotation column, in which control treatments were entered as “none”.

```

> GEO6907.list <- splitPerturbations( GEO6907.eSet,
+   control = "none",
+   factor.of.interest = "COMPOUND",
+   controlled.factors = "none"
+ )

```

The `annotate_eset_list` function collects the sample annotations from each individual treatment instances into a single data frame.

```

> sample.anno <- annotate_eset_list( GEO6907.list )
> head( sample.anno )

```

	Factor.Value..Dose.	Factor.Value..COMPOUND.
1	10	2,3-dimethoxy-1,4-naphthoquinone
2	20	Mercury (II) chloride
3	2	Cadmium chloride 2.5-hydrate
4	6.5	Nickel (II) chloride hexahydrate
5	200	Bis [(+)-tartrato] diantimonate (III) dipotassium trihydrate
6	20	Potassium dichromate

Next, we can perform a differential expression analysis separately for each treatment instance and collect all results in an `NChannelSet`. By providing a path to the `big.matrix` parameter, we instruct *gCMAP* to leverage the *bigmemory* package to store the assayData as binary files on disk. Next time the object is loaded, only a small description file will reside in memory. Subsets of the gene expression scores will be retrieved on demand, drastically reducing the amount of memory required to query multiple connectivity maps simultaneously.

```

> GEO6907.cmap <- generate_gCMAP_NChannelSet(
+   GEO6907.list,
+   sample.annotation = sample.anno,
+   big.matrix = file.path( tempdir(), "GEO6907.cmap" )
+ )

```

To see how many genes were up- and down-regulated in each experiment we apply a threshold to the z-score channel.

```

> GEO6907.sets <- induceCMAPCollection( GEO6907.cmap,
+   lower = -3, higher = 3,
+   element = "z" )
> setSize( GEO6907.sets )

```

	n.up	n.down	n.total
1	94	1345	1439
2	115	1043	1158
3	242	320	562
4	464	776	1240
5	151	1974	2125
6	146	1169	1315
7	1	165	166
8	1014	405	1419
9	724	187	911

## 2.3 Obtaining public gene set annotations

WikiPathways is an open, public platform dedicated to the curation of biological pathways. The `wiki2cmap` function automatically retrieves the latest pathway annotations from the WikiPathways website. It returns a `CMAPCollection` object, which represents the gene sets in a sparse matrix format. Sparse matrices requires less memory, but still allow us to use matrix algebra operations to speed up set-based calculations.

```
> wiki.hs <- wiki2cmap( species = "Homo sapiens",
+                      annotation.package = "org.Hs.eg.db" )
> save( wiki.hs, file = file.path( tempdir(), "wiki.hs.rdata" ) )
```

Similarly, the `KEGG2cmap` function generates a species-specific gene set collection with Entrez gene identifiers from the latest public release of the KEGG database and returns a `CMAPCollection` object. (The species identifier for *Homo sapiens* used by KEGG is "hsa".) This function requires the `KEGG.db` annotation package to be installed on your system.

```
> KEGG.hs <- KEGG2cmap( species = "hsa",
+                      annotation.package = "org.Hs.eg.db" )
> save( KEGG.hs, file = file.path( tempdir(), "KEGG.hs.rdata" ) )
```

## 2.4 Starting the *gCMAPWeb* application

To register reference datasets with the *gCMAPWeb* application, provide basic information about them in a nested list. (Consult the *gCMAP* package vignette for a detailed description of the configuration options.)

```
> config <- list(
+   species = list(
+     human = list(
+       annotation = "org.Hs.eg",
+       cmaps = list(
+         GEO6907 = file.path( tempdir(), "GEO6907.cmap.rdata" ),
+         Wiki = file.path( tempdir(), "wiki.hs.rdata" ),
+         KEGG = file.path( tempdir(), "KEGG.hs.rdata" )
+       )
+     )
+   )
+ )
> writeLines( as.yaml( config ), file.path( tempdir(), "gcmmap.yml" ) )
```

Afterward, we save this configuration file in yaml format to disk. (In this document, all objects are saved in the session-specific temporary directory, but they can reside anywhere on the system.) Now, start *gCMAPWeb* by providing the path to the configuration file.

```
> gCMAPWeb( config.file.path = file.path( tempdir(), "gcmmap.yml" ) )
```

The *gCMAPWeb* application will open in your default browser, ready for queries with gene sets of your choice.

### 3 RNAseq analysis of the benzo[a]pyrene response in HepG2 cells

In this example, we demonstrate *gCMAP*'s ability to process count data using the *DESeq* Bioconductor package. In 2012, van Delft et al. studied the response of human HepG2 cells to treatment with benzo[a]pyren, a potent carcinogen, using transcriptome sequencing (RNAseq) [11]. Global expression changes were profiled 12 and 24 hours after treatment. The authors deposited the raw sequencing reads at the European Nucleotide Archive (ENA) and also included the study in ArrayExpress. (PubMed accession 22889811, ArrayExpress accession E-GEOD-36242, ENA accession SRP011233.) For this example, we used the Bioconductor *HTSeqGenie* package to realign the raw reads to the human genome (hg19) to obtain the number of uniquely mapped reads for every protein coding gene. For convenience, the count matrix is included in with this article as Supplementary Data 2.

#### 3.1 Differential expression analysis of RNAseq data

```
> library( gCMAPWeb )
> library( DESeq )
```

To run this example, download the `Supplementary_data_2.txt` file accompanying this article and provide the full path to it to the `read.delim` function shown below.

```
> GSE36242.counts <- read.delim( "Supplementary_data_2.txt",
+                               row.names = "EntrezId" )
```

Next, we retrieve the sample annotations from ArrayExpress and extract the experimental factor of interest.

```
> accession <- "E-GEOD-36242"
> url <- paste( "http://www.ebi.ac.uk/arrayexpress/files/",
+              accession, "/",
+              accession, ".sdrf.txt",
+              sep="" )
> sample.anno <- read.delim(url, as.is=TRUE)
```

We subset the sample annotation table to the relevant rows and columns and reorder it to match the order of the count matrix columns.

```
> sample.anno <- sample.anno[seq(1, nrow(sample.anno), 2) ,c(30,33,36)]
> row.names( sample.anno ) <- sample.anno[, "Comment..ENA_RUN."]
> sample.anno[, "Comment..ENA_RUN."] <- NULL
> sample.anno <- sample.anno[ colnames( GSE36242.counts), ]
> colnames( sample.anno ) <- c("COMPOUND", "TIME")
> sample.anno
```

```
>
>          COMPOUND TIME
> SRR427095 Benzo[a]pyrene 12h
> SRR427096 Benzo[a]pyrene 12h
> SRR427097 Benzo[a]pyrene 24h
> SRR427098 Benzo[a]pyrene 24h
> SRR427099          DMSO 12h
> SRR427100          DMSO 12h
> SRR427101          DMSO 24h
> SRR427102          DMSO 24h
```



Now, the count matrix and the sample annotation data frame are combined into a `countDataSet` object, suitable as input for *gCMAP*.

```
> cds <- newCountDataSet( GSE36242.counts, conditions=sample.anno)
```

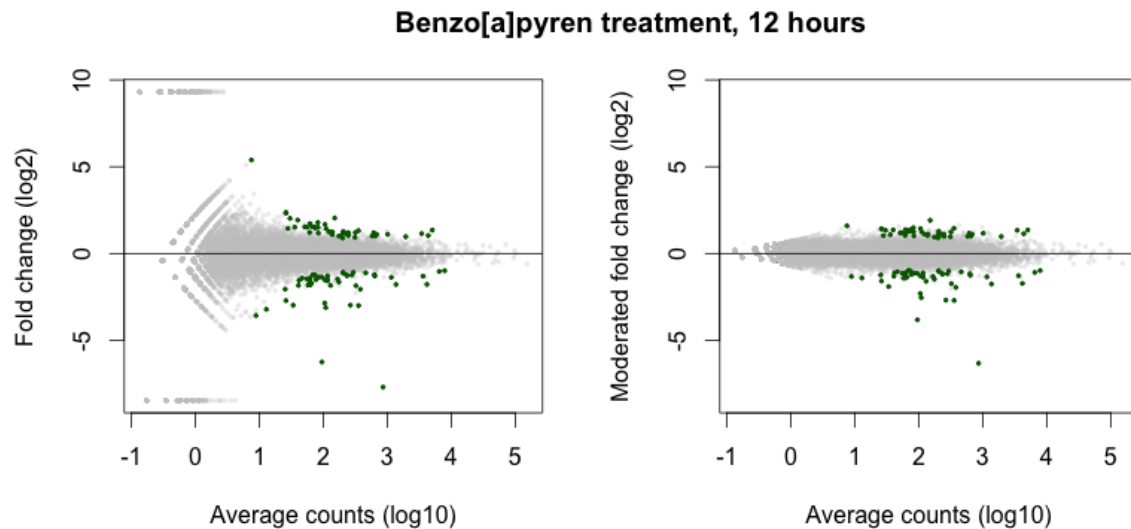
Next, we match the different treatment and control samples using the sample annotation provided by *ArrayExpress*, creating separate `CounDataSets` for each individual experimental condition.

```
> cds.list <- splitPerturbations( cds,
+                               control="DMSO",
+                               controlled.factors ="TIME",
+                               factor.of.interest ="COMPOUND")
```

We are now ready to perform a differential expression analysis for each perturbation, using *DESeq*'s `nbinomTest` function, and collect all results in the *GSE36242 NChannelSet*.

```
> GSE36242.cmap <- generate_gCMAP_NChannelSet( cds.list,
+                                             uids=c("24h", "12h"))
```

The `NChannelSet` contains five channels: the average counts across all experimental instances (`exprs`), the  $\log_2$  fold change calculated from normalized counts (`log_fc`), the  $\log_2$  fold change calculated after performing variance-stabilizing transformation of the counts (`mod_fc`), the raw p-value (`p`) and z-score obtained by transforming the raw p-value using a standard normal distribution (`z`).



**Figure 5:** Fold change versus count (MA) plots showing the effect of treating HepG2 cells with benzo[a]pyrene for 12 hours. On the left, the y-axis corresponds to  $\log_2$  fold change calculated from normalized counts. On the right, counts were transformed using a *DESeq*'s `varianceStabilizingTransformation` function before calculating the  $\log_2$  fold changes. Genes with a z-score  $>3$  or  $<-3$  are indicated in green.

```
> MA.plot <- function( A, M, z, ylab, xlab, alpha=0.25,
+                      ylim=c(-5,5), main=""){
+   plot( A,
```

```

+       M,
+       ylim=ylim,
+       xlab=xlab,
+       ylab=ylab,
+       main=main,
+       col=rgb(0.8,0.8,0.8,alpha=alpha), pch=20, cex=0.5)
+ points(A[abs(z) > 3], M[abs(z) >3], col="darkgreen", pch=20,
+        cex=0.5)
+ abline(h=0)
+ }
> A <- log10( assayDataElement( GSE36242.cmap, "exprs"), "12h" )
> z <- assayDataElement( GSE36242.cmap, "z"), "12h"
> ## Infinite log fold changes are hart to plot, so we set them
> ## to 110% of the largest observed finite value
> M <- assayDataElement( GSE36242.cmap, "log_fc"), "12h"
> M[ M == -Inf ] <- -1.1 * max( abs( M[ is.finite(M)] ))
> M[ M == Inf ] <- 1.1 * max( abs( M[ is.finite(M)] ))
> M.mod <- assayDataElement( GSE36242.cmap, "mod_fc"), "12h"
> par(mfrow=c(1,2))
> MA.plot( A, M, z,
+         ylab="Fold change (log2)",
+         xlab="Average counts (log10)",
+         ylim=range( M, na.rm=TRUE ))
> MA.plot( A, M.mod, z,
+         xlab="Average counts (log10)",
+         ylab="Moderated fold change (log2)",
+         ylim=range( M, na.rm=TRUE ))
> par(mfrow=c(1,1))
> title(main="Benzo[a]pyren treatment, 12 hours")

```

To see how many genes were up- and down-regulated, respectively, we apply a threshold to the z-score channel.

```

> GSE36242.sets <- induceCMAPCollection( GSE36242.cmap, element="z",
+                                       lower=-3, higher=3)
> setSizes( GSE36242.sets)

>      n.up n.down n.total
> 24h  456   340   796
> 12h   45    50    95

```

While only 95 genes show significant signs of differential expression after exposure to benzo[a]pyrene for 12 hours (figure 5), nearly 800 genes are affected after 24 hours. Next, we can use *gCMAP*'s gene set enrichment analysis methods to learn more about the observed drug response.

### 3.2 Connectivity mapping with *gCMAPWeb*

The connectivity map approach is aimed at identifying those experimental instances in the collection of reference experiments with significant similarity to the query [5]. Different similarity metrics and associated statistical tests have been proposed in the literature, several of which are available as methods in the *gCMAPWeb* package for interactive use.

For the *gCMAPWeb* application, we operationalized the connectivity mapping process by employing either Fisher's exact test (for queries with gene set identifiers) [1] or the JG summary score [3] ( for

directional gene set and profile queries). Both metrics can be computed without requiring time-consuming permutations of the reference database and their p-values can be estimated parametrically.

In this section, we will use the *gCMAPWeb* application to explore the benzo[a]pyrene response in HepG2 cells. Like the authors of the original study, we take advantage of pathway annotations compiled by the WikiPathways project.

As a second source of pathway annotations, we will compile gene sets from the reactome database with the `reactome2cmap` command. *gCMAP* also provides an analogous `go2cmap` function to leverage data from the *GO.db* Gene Ontology annotation package. These functions require the respective Bioconductor packages to be available on your system.

```
> reactome.hs <- reactome2cmap( species="Homo sapiens",
+                             annotation.package="org.Hs.eg.db")
> save( reactome.hs, file=file.path( tempdir(), "reactome.hs.rdata"))
```

Now, we are ready to deploy the *gCMAPWeb* application by exporting the paths to the reference datasets to a configuration file. As a control, we will also include the `NChannelSet` with the results from the benzo[a]pyrene RNAseq analysis itself as a connectivity map.

```
> library(yaml)
> config <- list(
+   species=list(
+     human=list(
+       annotation="org.Hs.eg",
+       cmaps=list(
+         WikiPathways=file.path( tempdir(), "wiki.hs.rdata"),
+         Reactome=file.path( tempdir(), "reactome.hs.rdata"),
+         Benzopyrene=file.path( tempdir(), "GSE36242.cmap.rdata")
+       )
+     )
+   )
+ )
> writeLines(as.yaml(config),file.path( tempdir(), "gcmmap.yaml"))
```

We can start the web application by providing the path to the configuration file.

```
> library(gCMAPWeb)
> gCMAPWeb(config.file.path = file.path( tempdir(), "gcmmap.yaml"))
```

*gCMAPWeb* offers three different query options, addressing the following questions:

- **Non-directional query:** Is there significant overlap between your query genes and a reference gene set?
- **Directional query:** Do your query genes consistently change expression in other experiments?
- **Profile query:** Are genes changing expression in other studies consistently up- or down-regulated in your experiment?

Note that additional methods to process complete, replicated microarray experiments via sample permutation and rotation approaches are available in the *gCMAP* command line package.

**Figure 6:** Screenshot of the *gCMAPWeb* submission page for non-directional queries.

### 3.2.1 Performing a non-directional query

To identify significant overlaps between genes responding to benzop[a]yrene treatment after 12 hours, we perform a **non-directional query**. We retrieve the identifiers of all differentially regulated genes from the GSE36242.sets object with the `geneIds` command.

```
> library(gCMAP)
> cat( paste( geneIds(GSE36242.sets)[["12h"]], collapse=", ") )
```

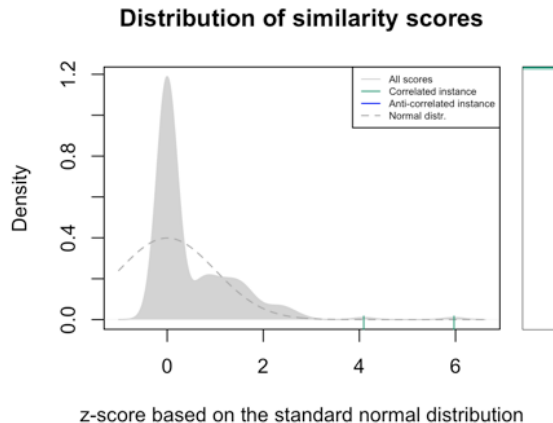
set	trend	FDR	effect	LOR	nSet	Genes	Image
Hs_Cholesterol_Biosynthesis_WP197_44991	over	4.56e-07	5.97	4.36	17	6	
Hs_Benzo[a]pyrene_metabolism_WP696_41182	over	4.01e-03	4.09	4.18	9	3	

**Figure 7:** Screenshot of the *gCMAPWeb* result table for WikiPathway genes significantly overrepresented in genes differentially regulated after 12 hours of benzop[a]yrene treatment.

We ignore the fact that some genes are up- and others downregulated (for now) and paste the list into the textbox of the *gCMAPWeb* **Non-directional query** submission form (Figure 6).

By default, *gCMAPWeb* employs Fisher's exact test [1] for set-wise comparisons, which tests for significant over-/under-representation of the query genes in other reference gene sets. This test requires only a list of gene identifiers, but it assumes statistical independence between genes (figure 9). For a detailed discussion of gene-set enrichment approaches, please refer to Goeman and Bühlmann [2].

On the *gCMAPWeb* output page, results are presented in separate tabs for each reference database:



**Figure 8:** Screenshot of the *gCMAPWeb* density plot showing the distribution z-scores based on a standard normal distribution for all WikiPathway categories. Gene sets significantly overrepresented in genes differentially regulated after 12 hours of benzop[a]yrene treatment are indicated in the rug. For reference, a standard normal distribution is shown (dashed line).

Querying the WikiPathway database returns two pathways with significant p-values after correcting for multiple testing (figure 7): Cholesterol biosynthesis and benzop[a]yrene metabolism. (Both pathways are also significantly enriched after 24 hours of treatment, as will be highlighted on page 13).

*gCMAPWeb* also displays the results of the gene-set enrichment analysis in graphical form, as a density plot of the distribution of similarity scores across all queried gene sets (figure 8). To allow for comparison between different gene-set enrichment methods, p-values are transformed to z-scores based on a standard normal distribution. In this analysis, most gene sets receive z-scores close to zero, while the two significantly overrepresented categories appear as outliers with positive scores.

For each significantly enriched gene set, detailed information about member genes is available via the link in the **Genes** column of the result table. For categorial comparisons, a pie chart is produced, highlighting the fraction of overlap between query and reference set.

The second panel presents results from querying the Reactome gene set collection, again highlighting the effect of benzo[a]pyrene on the cholesterol metabolism as well as the regulation of gene expression.

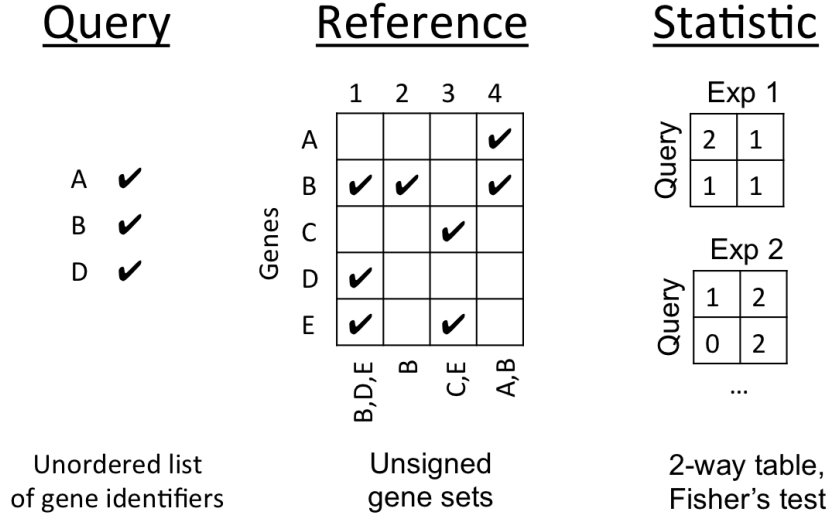
Finally, the third panel contains results from querying the Benzo[a]pyrene dataset itself. To enable set-wise comparisons, *gCMAPWeb* automatically applied a z-score threshold (customizable via the global `lower.threshold` and `higher.threshold` parameters, see package vignette for details) to derive gene sets from the experiments. Our query, consisting of genes significantly regulated at 12 hours, yields perfect overlap with the experiment it was derived from and also shows significant overlap with genes differentially expressed at the later timepoint. As quantitative information is available in the reference database, *gCMAPWeb* displays a heatmap with differential expression scores for our queries found in the two experiments (figure 10).

The other two query types, **directional** and **profile** queries, take advantage of the quantitative information available in *gCMAP* connectivity maps. They will be explored in section 4 below.

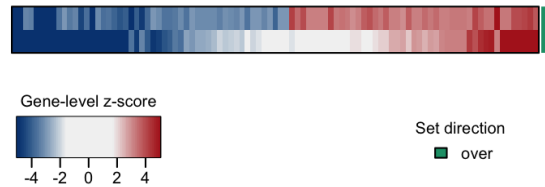
### 3.2.2 Connectivity mapping in the command line

All methods and classes used by the *gCMAPWeb* application are also available in the command line. The following examples reproduce the queries executed above by calling the corresponding *gCMAP* methods directly.

As a first analysis, we used Fisher’s exact test to check for overlap between genes significantly changing their expression upon exposure to benzo[a]pyrene with gene set annotated in the WikiPathways



**Figure 9:** Schematic overview of a non-directional query. The query gene set is an unordered list of gene identifiers. The reference connectivity map can either be a collection of gene annotation categories (e.g., from WikiPathways, KEGG, etc.) or contain quantitative differential expression scores. In the latter case, a significance threshold is applied to define regulated gene sets on the fly. For each reference gene set, a two-way table is created, summarizing the overlap between query and reference and a p-value is obtained using Fisher's exact test.



**Figure 10:** Heatmap produced by *gCMAPWeb* when quantitative scores are available in the reference database. Rows correspond to experimental instances in the reference, columns correspond to query genes in the order they were submitted. Differential expression scores observed in the reference experiment are indicated from blue (down-regulated) to red (up-regulated). The green annotation bar on the right indicates significant overlap of query and reference gene sets.

database.

```
> fisher.results <- fisher_score( GSE36242.sets, wiki.hs,
+                               universe=featureNames(GSE36242.sets)
+                               )
> cmapTable( fisher.results[["12h"]], n=5)[,c(1,4,6:8)]
> cmapTable( fisher.results[["24h"]], n=5)[,c(1,4,6:8)]
```

All *gCMAP* gene-set enrichment methods return an object of class `RclassCMAPResults`, containing the information about the statistical test used, one or more summary statistics, instance annotations from the reference database and (optionally) gene-level scores for each tested gene set. The `cmapTable` function coerces the results into a standard data frame.

As reported by van Delft and co-workers [11], genes involved in benzo(a)pyrene metabolism, the Keap1-Nrf2 pathway [8] and cholesterol biosynthesis are significantly overrepresented, both after 12 and 24 hours.

While Fisher's test only considered the overlap between groups of gene identifiers, the *gCMAP* package also offers methods that take into account the strength of the observed expression changes. For example, the JG score summarizes the observed z-scores and assigns a p-value based on a standard normal distribution [3]. (For more on the JG score and directional gene queries see section 4.)

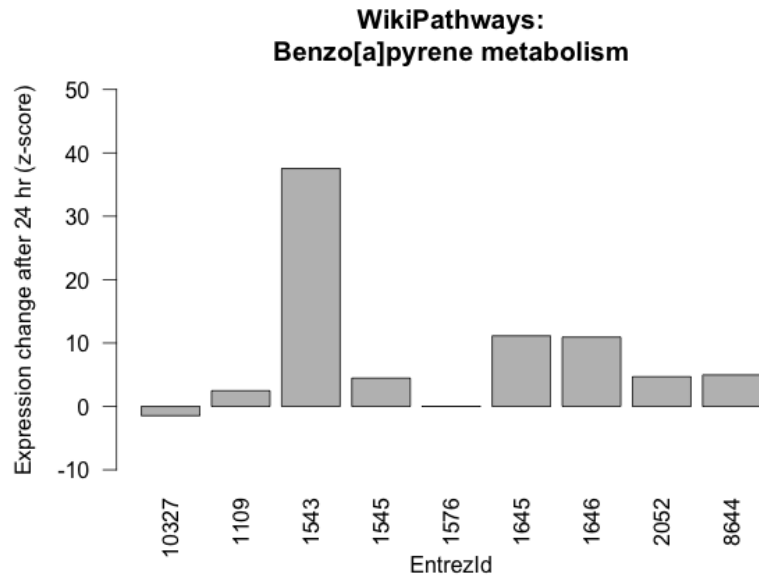
```
> res <- gsealm_jg_score( GSE36242.cmap, wiki.hs, keep.scores=TRUE)
> cmapTable( res[["12h"]], n=5)[,c(1,4,5)]
```

	set	padj	effect
1	Hs_Benzo(a)pyrene_metabolism_WP696_41182	1.735171e-22	10.281400
2	Hs_Cholesterol_biosynthesis_WP197_44991	7.701360e-17	-8.865919
3	Hs_Electron_Transport_Chain_WP111_41171	4.872039e-13	7.780917
4	Hs_Proteasome_Degradation_WP183_59174	1.995547e-12	7.563220
5	Hs_Oxidative_phosphorylation_WP623_45305	3.579053e-08	6.129503

The sign of the effect size (JG score) column is positive for up- and negative for down-regulated sets, e.g., for the Benzo[a]pyrene metabolism and cholesterol biosynthesis pathways, respectively.

As we set the `keep.scores` parameter of the `gsealm_jg_score` function to `TRUE`, the gene-level scores are stored in the `res` `CMapResults` object. (We could also retrieve them directly from the original reference dataset).

```
> scores.24h <- geneScores( res[["24h"]][1,] )
> barplot( unlist( scores.24h ), names.arg=names( scores.24h[[1]] ),
+         las=2, ylab="Expression change after 24 hr (z-score)",
+         xlab="EntrezId",
+         main="WikiPathways:\nBenzo[a]pyrene metabolism",
+         ylim=c(-10,50))
```



**Figure 11:** Differential expression z-scores after treatment of HepG2 cells for 24 hours for genes annotated in the Benzo[a]pyrene metabolism pathway by WikiPathways.

## 4 Building and querying the Broad connectivity Map

In 2006, Lamb and co-workers recorded the response of five human cell lines to hundreds of chemical compounds to generate the Broad connectivity Map [5]. The results for the first phase of this endeavor, release 1 of the data, is available in the ArrayExpress repository. (PubMed accession 17008526, ArrayExpress accession E-GEOD-5258.)

We can use the `ArrayExpress` function from the *ArrayExpress* Bioconductor package to download the raw data. **Note:** ca. **2Gb** of data will be retrieved.

```
> GEOD5258.batch <- ArrayExpress( "E-GEOD-5258" )
```

Since its deposition, the first array platform has changed its name, so we update its annotation string.

```
> annotation(GEOD5258.batch[[1]]) <- "hthgu133a"
```

As this experiment was performed on two different array platforms, `ArrayExpress` returns a list with two `affyBatch` objects, one for each array platform. We normalize each object separately. (The necessary annotation packages (*hthgu133a.db* and *hgu133a.db*) are available from Bioconductor.)

```
> library("hthgu133a.db")
> library("hgu133a.db")
> GEOD5258.rma <- lapply( GEOD5258.batch, rma )
> rm( GEOD5258.batch )
```

We map probe IDs to Entrez identifiers and average data for genes assayed by multiple probes.

```
> GEOD5258.eSets <- lapply( GEOD5258.rma, mapNmerge )
> rm( GEOD5258.rma )
```

Now the two normalized `ExpressionSets` can be combined into one.

```
> GEOD5258.eSet <- mergeCMAPs( GEOD5258.eSets[[1]], GEOD5258.eSets[[2]] )
> rm( GEOD5258.eSets )
```

Next, we identify the experimental factors of interest from the sample annotations provided by `ArrayExpress` and shorten them to make them easier to read.

```
> conditions <- grep("^Factor", varLabels( GEOD5258.eSet ), value=TRUE)
> pData( GEOD5258.eSet ) <- pData( GEOD5258.eSet )[, conditions]
> varLabels( GEOD5258.eSet ) <- c("CellLine", "Vehicle",
+                               "Compound", "Time", "Dose")
> pData( GEOD5258.eSet )[30,]
```

	CellLine	Vehicle	Compound	Time	Dose
GSM119261	MCF7	DMSO	(-)-catechin	6	1.1e-05

This preprocessed `ExpressionSet` is now suitable as input for *gCMAP*.

### 4.1 Differential expression analysis

The `splitPerturbations` function automatically combines matched treatment and control samples into separate `ExpressionSets`, one for each tested condition, and returns them in a list. We are interested in studying the effect of the different `Compounds`. Controls received treatment “none” and need to be matched to perturbations performed in the same `CellLine`, treated with the correct `Vehicle` and for the same amount of `Time`.



```

> GEOD5258.list <- splitPerturbations( GEOD5258.eSet,
+                                     factor.of.interest="Compound",
+                                     control="none",
+                                     controlled.factors=c("CellLine",
+                                                         "Vehicle",
+                                                         "Time")
+                                     )
> sample.anno <- annotate_eset_list( GEOD5258.list )
> sample.anno[23,]

```

```

      CellLine Vehicle          Compound Time Dose
23      MCF7      DMSO 15-delta prostaglandin J2    6 1e-05

```

We obtain a list with 281 treatment conditions with biological replication, suitable for differential expression analysis. Again, we use the `generate_gCMAP_NChannelSet` function to analyze all instances (using *limma*).

```

> GEOD5258.cmap <- generate_gCMAP_NChannelSet( GEOD5258.list,
+                                             uids=paste( "Exp",
+                                                         1:length( GEOD5258.list ), sep=""),
+                                             big.matrix=file.path( tempdir(),
+                                                                     "GEOD5258.cmap"),
+                                             sample.annotation=sample.anno
+                                             )
> assayDataElementNames( GEOD5258.cmap )

```

```
[1] "exprs" "log_fc" "p" "z"
```

By thresholding the z-scores we identify significantly up- and down-regulated genes and store them in a `CMAPCollection`. (Here, a z-score cutoff of  $>3$  or  $<-3$  is chosen, as the p-values stored in the `NChannelSet` have not been corrected for multiple testing. If adjusted p-values are desired, simply apply the `p.adjust` function to the `pval` element of the `NChannelSet`.)

```

> GEOD5258.sets <- induceCMAPCollection( GEOD5258.cmap,
+                                       element="z",
+                                       higher=3,
+                                       lower=-3 )
> GEOD5258.sets <- minSetSize( GEOD5258.sets, min.members=10 )
> head( setSizes( GEOD5258.sets ) )

```

	n.up	n.down	n.total
Exp1	51	21	72
Exp2	2	39	41
Exp7	0	14	14
Exp8	13	285	298
Exp9	0	71	71
Exp11	1	400	401

## 4.2 Querying the Broad connectivity map with *gCMAPWeb*

To register the Broad connectivity map with the *gCMAPWeb* application, we provide basic information in a nested list and save it in yaml format.

```

> config <- list(
+   species=list(
+     human=list(
+       annotation="org.Hs.eg",
+       cmaps=list(
+         Broad=file.path( tempdir(), "GEOD5258.cmap.rdata" )
+       )
+     )
+   )
+ )
> writeLines(as.yaml(config),file.path( tempdir(), "gcmmap.yml"))

```

We can start the web application by providing the path to the configuration file.

```

> gCMAPWeb(config.file.path = file.path( tempdir(), "gcmmap.yml"))

```

#### 4.2.1 Non-directional query: assessing overlap between query and reference datasets

As discussed on page 12, we can perform a non-directional test by retrieving the identifiers of all differentially regulated genes from the `CMAPCollection` object with the `geneIds` method and paste them into `gCMAPWeb`'s non-directional query submission form.

```

> cat(paste(geneIds( GEOD5258.sets[,"Exp23"]), sep=" "))

1026 10420 10723 10849 10912 124222 124565 133 1466 1605 1645 1646 1649 1663
1843 1999 206358 2114 23175 23338 2355 23598 23645 23657 23729 23764 2534
25803 25837 25888 26136 2669 2703 27289 29948 3162 3310 3337 374655 375449
4097 4131 467 5029 51447 54455 54541 54894 54910 55122 55290 55323 55652
55893 57016 57493 58190 6464 64782 6509 7043 7277 7422 7841 79094 79803
80271 80328 8061 81621 8419 84705 8744 8795 8820 8878 899 9020 90627 9203
9283 9682 9903

```

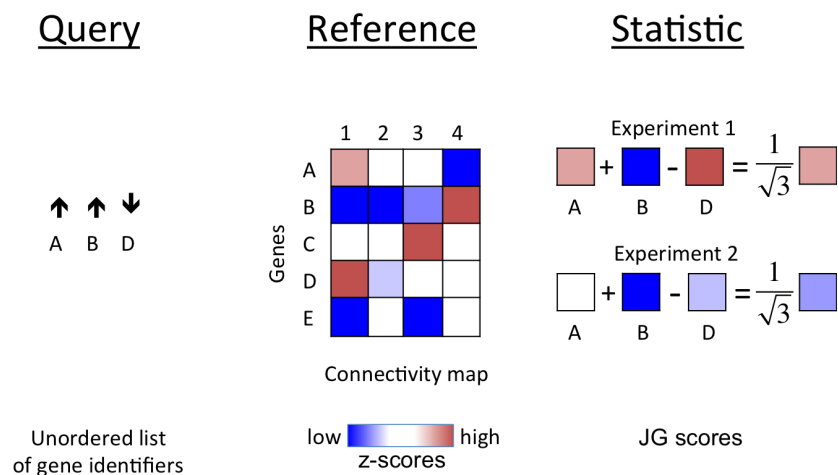
As expected, the query instance itself — treatment of MCF7 cells with 15-delta prostaglandin J2 — is returned as the top hit, with multiple other reference sets also displaying highly significant overlap with the query. (Please refer to page 13 for details, where this query is executed in the command line. )

#### 4.2.2 Directional query: using information about the direction of gene expression changes

While a non-directional query merely detects overrepresentation between different gene lists, a **directional query** can take advantage of the quantitative information stored in the connectivity map to see if the query genes consistently change expression in other experiments.

By default `gCMAPWeb` uses the JG score to summarize the differential expression scores across all gene set members in a reference dataset. As the differential expression scores of up- and down-regulated genes have the opposite sign, it is important to include information about the direction of gene expression change in the query. (Otherwise, scores for up- and down-regulated gene set members would cancel each other out.)

As an example for a directional query, we will use one drug perturbation from the Broad connectivity map to query the full reference dataset. In experiment 23, MCF7 cells were treated with 15-delta prostaglandin J2, an inhibitor of NF  $\kappa$  B signaling [10]. We apply a z-score threshold to the Connectivity map and retrieve the identifiers for up- and down-regulated genes specifically for experiment 23. To identify perturbations that led to similar expression changes, we paste the identifiers into “up-” and “down-regulated query” text fields of the `gCMAPWeb` **directional query** submission form (figure 13).



**Figure 12:** Schematic overview of the JG score calculation. Query gene identifiers are accompanied by a sign vector (up-/down-regulated) (left). For each experiment in the reference connectivity map, the scores for all query genes are retrieved; scores for up-regulated genes are added while those for down-regulated genes are subtracted, thereby preserving the directional contribution of all genes. Finally, to normalize for differing gene set size, the total score is multiplied by the square root of the number of genes in the set.

```
> pData( GEOD5258.sets )["Exp23",c(2:6)]

      CellLine Vehicle      Compound Time Dose
Exp23   MCF7     DMSO 15-delta prostaglandin J2  6 1e-05

> cat( "Up-regulated" )
> cat( paste( upIds( GEOD5258.sets )["Exp23"], sep="," ) )
> cat( "Down-regulated" )
> cat( paste( downIds( GEOD5258.sets )["Exp23"], sep="," ) )

Up-regulated
1026 10723 10912 133 1466 1645 1646 1649 1843 206358 2114 23175 2355 23645
23657 23764 2534 25888 26136 2669 27289 29948 3162 3310 3337 4097 4131 467
51447 54455 54541 55122 55290 55323 55652 57016 57493 6464 64782 6509 7277
7422 79094 80271 80328 8061 8744 8795 8878 9020 9682 9903

Down-regulated
10420 10849 124222 124565 1605 1663 1999 23338 23598 23729 25803 25837
2703 374655 375449 5029 54894 54910 55893 58190 7043 7841 79803 81621 8419
84705 8820 899 90627 9203 9283
```

On the result page, *gCMAPWeb* offers two overview plots (figure 14). On the left, a density plot shows the distribution of JG scores obtained for all experiments in the connectivity map. Genes with significantly positive or negative scores are indicated in the rug in green (positive, correlated) or blue (negative, anti-correlated) dashes, respectively.

In this example, the highest similarity score (JG score >38) is returned for the query instance itself. In addition, several other drug perturbations in the same cell line achieve highly positive scores, including Z-Leu-Leu-Leu-CHO (also known as MG-132) (JG score >30) and celastrol (>25).

Gene-level plots, displaying the differential expression scores for the individual query genes in the respective experiments, are available as thumbnails in the result table or on the gene result page, accessible via the hyperlink in the “Genes” column of the table.

**gCMAP** ↑

**Directional query: Please specify up- and/or down-regulated gene sets** More Info

The following **up-regulated** gene identifiers

1026 10723 10912 133 1466 1645 1646 1649 1843 206358 2114 23175 2355 23645  
 23657 23764 2534 25888 26136 2669 27289 29948 3162 3310 3337 4097 4131 467  
 51447 54455 54541 55122 55290 55323 55652 57016 57493 6464 64782 6509 7277  
 7422 79094 80271 80328 8061 8744 8795 8878 9020 9682 9903

no file selected

The following **down-regulated** gene identifiers

10420 10849 124222 124565 1605 1663 1999 23338 23598 23729 25803 25837 2703  
 374655 375449 5029 54894 54910 55893 58190 7043 7841 79803 81621 8419 84705  
 8820 899 90627 9203 9283

no file selected

are from and represent

human  Entrez identifiers

Gene symbols

**Please choose one or more reference databases**

Broad

**Figure 13:** Screenshot for the *gCMAPWeb* submission form for directional queries.

Both treatment with Z-Leu-Leu-Leu-CHO and celestrol lead to significant shifts in the expression of the query genes (figure 15). Interestingly, like 15-delta prostaglandin J2, both Z-Leu-Leu-Leu-CHO and celestrol are known inhibitors of NF $\kappa$ B signaling [7, 9]. It is tempting to speculate that the highly similar gene expression changes observed after treating MCF7 cells with these three compounds may be a result of this shared biomolecular mechanism of action.

### 4.2.3 Profile query: submitting differential expression scores

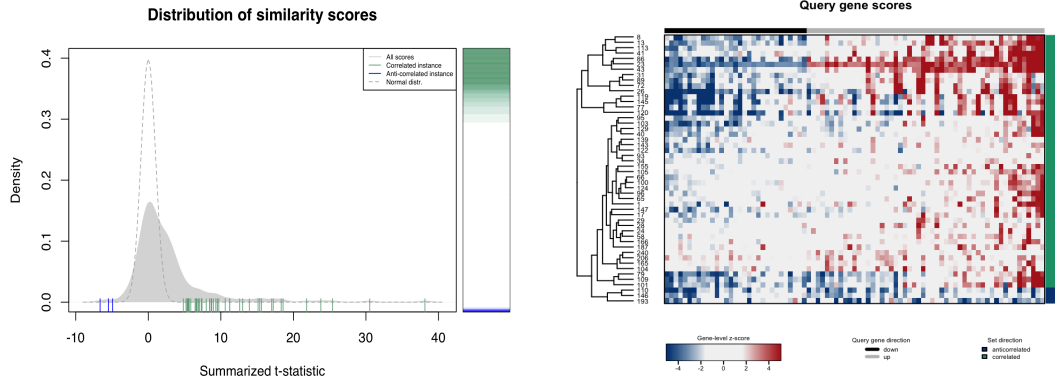
The third query type offered by *gCMAPWeb*, **profile query**, is closely related to the directional query option outlined above. Instead of applying a threshold to the query experiment, though, directional gene sets are defined for each experiment in the reference connectivity map (figure 16). Then, the JG score summary is obtained for each reference gene set by retrieving the scores from the submitted query scores. This allows users to assess whether gene sets defined in other studies are consistently up- or down-regulated in the query experiment.

We can obtain the full table of scores from the connectivity map:

```
> head(assayDataElement( GEOD5258.cmap, "z")[, "Exp23", drop=FALSE])
```

```

      Exp23
10    -1.12296718
100    0.09706779
1000  -0.39198825
10000  0.07649412
10001  0.42311730
10002  0.16007229
```

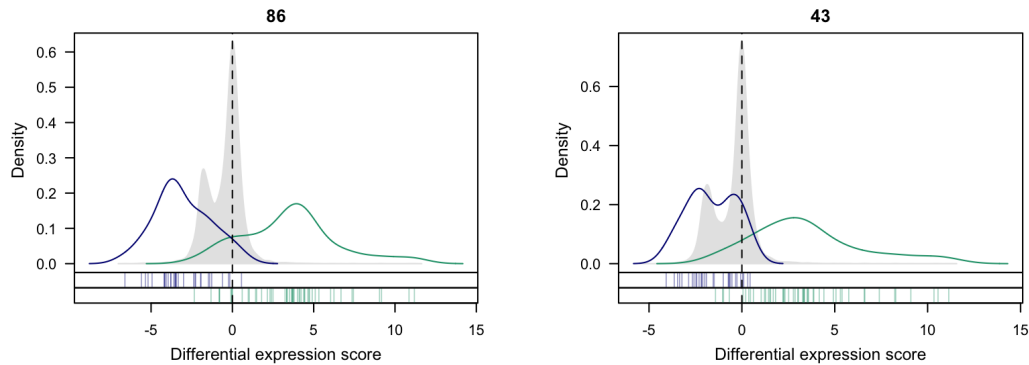


**Figure 14:** Left: Distribution of JG scores obtained for all experiments in the connectivity map. Genes with significantly positive or negative scores are indicated in the rug in green (positive, correlated) or blue (negative, anti-correlated) dashes, respectively. Right: Heatmap of gene-level scores for the top 50 significant gene sets. Gene sets are indicated as rows, query genes as columns. The column annotation bar shows whether genes were submitted as “up-regulated” (grey) or “down-regulated” queries. The row annotation bar indicates whether the expression of the query genes was correlated (green) or anti-correlated (blue) to that specified in the query.

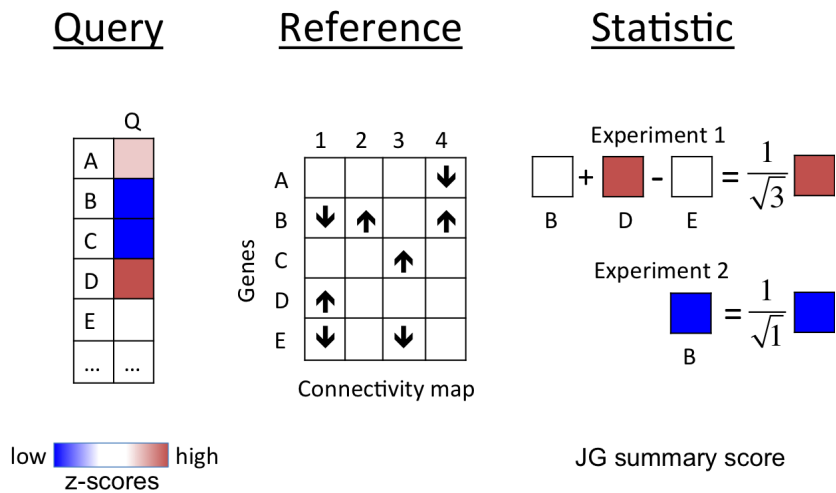
As this list of scores is very long (12701 genes) , it is more convenient to first write it to a text file and upload it into the *gCMAPWeb* interface directly (figure 17).

```
> write.table(assayDataElement( GEOD5258.cmap, "z")[, "Exp23", drop=FALSE],
+             file="z_scores.txt", quote=FALSE, row.names=TRUE,
+             col.names=FALSE, sep=",")
```

As expected, querying the Broad connectivity map with the z-scores observed for 15-delta prostaglandin J2 treatment of MCF7 cells returns the query instance itself as a top hit. The next best result, treatment of the same cells with 17-allylamino-geldanamycin is shown in figure 17. Genes up-regulated in response to 17-allylamino-geldanamycin also receive positive z-scores upon treatment with 15-delta prostaglandin J2, the query experiment. Also, genes down-regulated by 17-allylamino-geldanamycin have the tendency to be down-regulated in the query experiment, as indicated by the shift of blue density/rug to the left.




**Figure 15:** Differential gene expression z-scores obtained by the query genes in experimental treatments of MCF7 cells with Z-Leu-Leu-Leu-CHO (left) and celastrol (right). The filled grey density shows the distribution of all z-scores in the respective experiment. Query genes are indicated in the rug; genes submitted as “up-regulated” are shown in green, those submitted as “down-regulated” are indicated in blue.



**Figure 16:** Schematic overview of *gCMAPWeb*'s profile query type. Instead of applying a threshold to the query experiment (left), the reference connectivity map is converted into a collection of directional gene sets (center). Then, the JG score is calculated by summarizing the z-scores observed in the query experiment, separately for each of these reference gene sets.

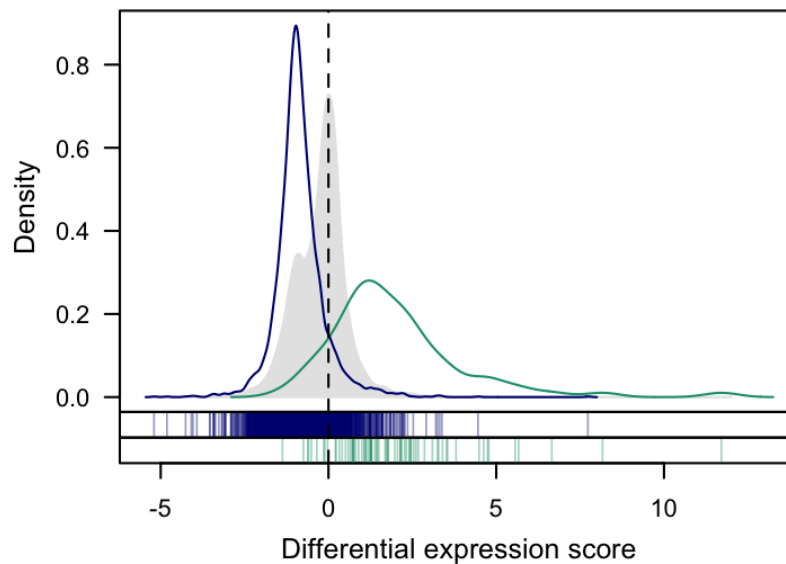
**Profile query: Please provide gene identifiers and expression scores**

The following gene identifiers and expression scores

 z\_scores.txt

are from  human and represent  Entrez identifiers  Gene symbols

**Figure 17:** Screenshot of the profile submission page. Gene identifiers and scores are uploaded as a csv file (red box).



**Figure 18:** *gCMAPWeb* density plot of gene scores after 15-delta prostaglandin J2 treatment of MCF7 cells. The filled grey density shows the distribution of scores for all genes on the array. The scores for genes observed as significantly up- or down-regulated in a different experiment, treatment of MCF7 cells with 17-allylamino-geldanamycin, are highlighted in green (up-regulated by 17-allylamino-geldanamycin) and blue (down-regulated by 17-allylamino-geldanamycin). Clearly, genes up-regulated in response to 17-allylamino-geldanamycin are also up-regulated in the query experiment, indicated by the strong shift of the green density/rug to positive z-scores.

## 4.3 Queries in the command line

### 4.3.1 Non-directional queries

To reproduce the results of a non-directional query with *gCMAPWeb*, the `fisher_score` method can be used. The following command evaluates the overlap of genes observed as significantly differentially expressed in MCF7 cells after treatment with 15-delta prostaglandin J2 (Exp23) with all other gene sets in the Broad connectivity map.

```
> res <- fisher_score( GEOD5258.sets["Exp23"], GEOD5258.sets,
+                     universe=featureNames( GEOD5258.sets))
> cmapTable( res )[1:3, c(1,2,4,6,9,11)]
```

	set	trend	padj	LOR	UID	Vehicle
1	Exp23	over	2.124065e-214	Inf	Exp23	DMSO
2	Exp86	over	1.275727e-74	5.095233	Exp86	DMSO
3	Exp43	over	9.724655e-48	4.802453	Exp43	DMSO

As observed above, the three NF $\kappa$ B inhibitors display the largest overlap. As expected, the query itself receives a perfect score — an infinite log odds ratio.

### 4.3.2 Directional queries

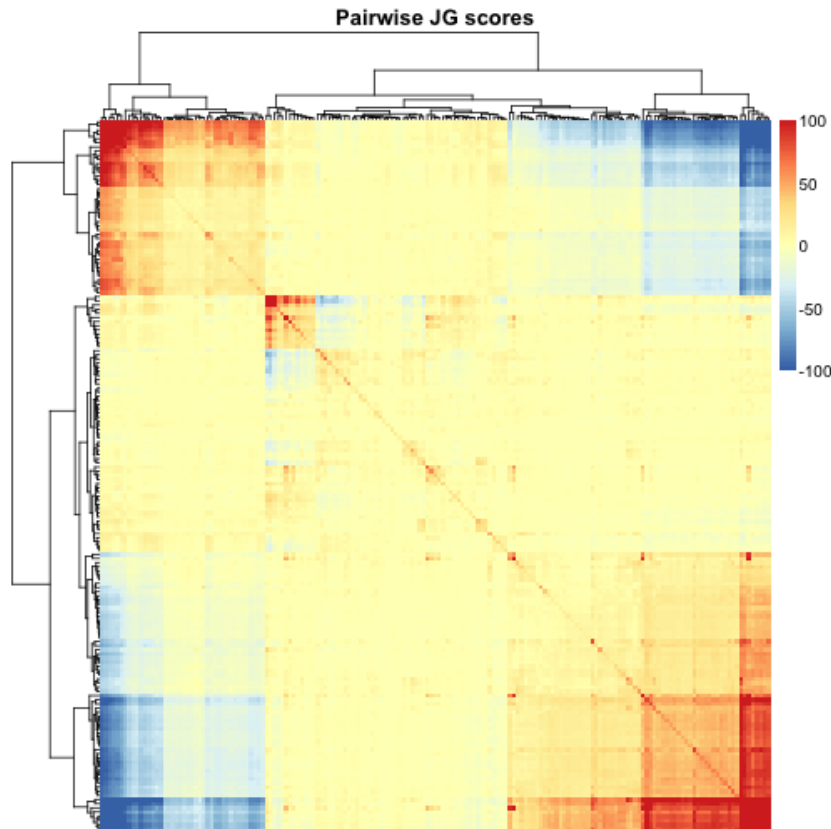
*gCMAP* offers efficient methods to perform large numbers of queries in the command line. For example, it is straightforward to calculate the pairwise Fisher or JG summary scores [3] between all gene sets in the Broad connectivity map.

```
> res.fisher <- fisher_score( GEOD5258.sets,
+                             GEOD5258.sets,
+                             universe=featureNames( GEOD5258.sets ))
> res <- gsealm_jg_score( GEOD5258.cmap[,sampleNames( GEOD5258.sets)],
+                         GEOD5258.sets )
```

By performing the all-versus-all comparisons, we obtained two scores for each pair of experimental conditions, with each instance used first as a query and then also as a reference. To obtain a symmetrical score we average the “query vs. reference” and “reference vs query” scores for each pair. Finally, we use the `pheatmap` function from the *pheatmap* Bioconductor package to cluster and display the results in a heatmap.

```
> library( pheatmap )
> scores <- sapply( res, function(x) {
+   effect(x)[ sampleNames( GEOD5258.sets) ]
+ })
> scores[is.nan(scores)] <- 0
> scores <- (scores + t( scores ) )/2
> scores[ abs( scores ) > 100] <- 100*sign(scores[ abs( scores ) > 100])
> pheatmap( scores,
+           breaks=seq(-100,100, length=101),
+           show_rownames=FALSE,
+           show_colnames=FALSE,
+           main="Pairwise JG scores",
+           border_color=NA
+           )
```





**Figure 19:** Heatmap of average pairwise JG scores for all Broad connectivity map (v1) experiments with at least ten significant genes.

## 5 Quality control

Like individual microarray experiments, reference instances compiled for use in a connectivity map must be of sufficient quality to provide useful information. As a quality control step, we routinely inspect the z-score density distributions and MA plots for connectivity map experiments.

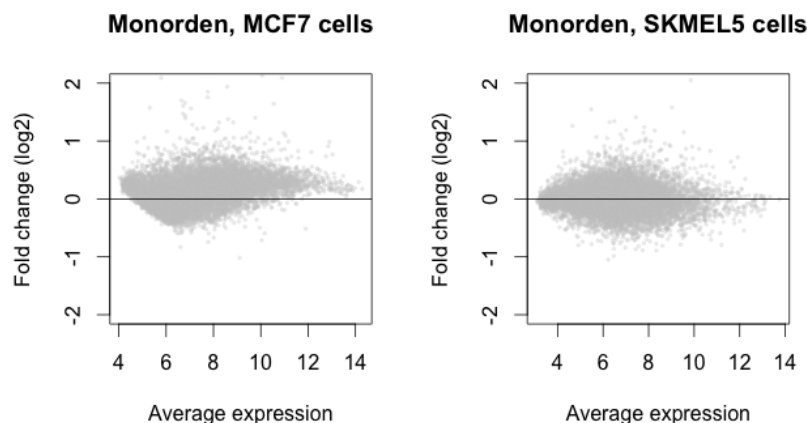
For example, the MA plot for treatment of MCF7 cells with Monorden for 6 hours reveals that the center of the log<sub>2</sub> fold change distribution is markedly up-shifted, highlighting potential normalization issues [6]. In contrast, the MA plot for the same treatment in SKMEL5 cells appears normal, with a log<sub>2</sub> fold change centered on zero (figure 20). As a consequence, the distribution of z-scores for genes in MCF7 cells is bi-modal, while the distribution for those in SKMEL5 cells is approximately normal (figure 21).

```
> pData( GEOD5258.cmap)[c(89,90),]
> par(mfrow=c(1,2))
> MA.plot(assayDataElement( GEOD5258.cmap, "exprs")[,89],
+         assayDataElement( GEOD5258.cmap, "log_fc")[,89],
+         NA,
+         main="Monorden, MCF7 cells",
+         xlab="Average expression",
+         ylab="Fold change (log2)",
+         ylim=c(-2,2))
```

```

> MA.plot(assayDataElement( GEOD5258.cmap, "exprs")[,90],
+         assayDataElement( GEOD5258.cmap, "log_fc")[,90],
+         NA,
+         main="Monorden, SKMEL5 cells",
+         xlab="Average expression",
+         ylab="Fold change (log2)",
+         ylim=c(-2,2))
> par(mfrow=c(1,1))

```



**Figure 20:** MA plot for MCF7 (left) and SKMEL5 (right) cells treated with Monorden for 6 hours. Genes with a z-score  $>3$  or  $<-3$  are indicated in green.

Skew or shifts in the z-score distribution can cause random sets of genes to receive high similarity scores. As a consequence, such reference experiments tend to be reported as matches to many different queries — spurious matches if the shift is due to failed normalization, or correct but hard to interpret matches if the treatment in the reference experience has caused significant changes to thousands of genes.

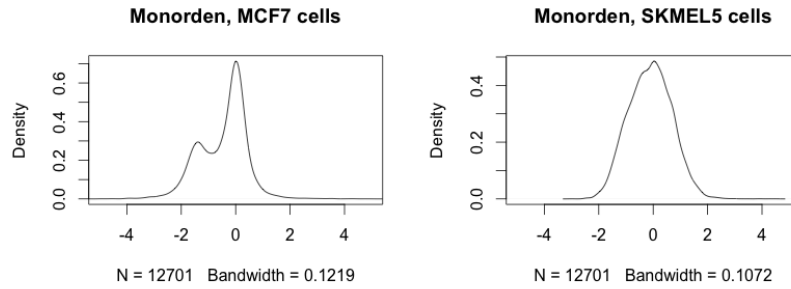
```

> par(mfrow=c(1,2))
> plot( density( assayDataElement( GEOD5258.cmap, "z")[,89]),
+       main="Monorden, MCF7 cells", xlim=c(-5,5))
> plot( density( assayDataElement( GEOD5258.cmap, "z")[,90]),
+       main="Monorden, SKMEL5 cells", xlim=c(-5,5))
> par(mfrow=c(1,1))

```

By default, the `generate_gCMAP_NChannelSet` function attempts to correct small global shifts in the z-score distribution by centering on zero. This is not sufficient, however, to address major normalization failures. (Please consult the man page for more details.) To identify problematic reference instances and flag them for removal, we routinely record the mode (before centering) and median absolute deviation (MAD) of the per-gene z-scores for each CMAP experiment. (Setting the `report.center` parameter of the `generate_gCMAP_NChannelSet` function to `TRUE` will report the `z.shift` and `z.mad` for each experiment in the `phenoData` slot of the returned `NChannelSet`.)

For example, experiment 26, treatment of MCF7 cells with allylamino-geldanamycin, received a score of 5 or above for nearly 60 percent of all queries. Inspection of the MA plot for this experiment (figure 22, left) shows that the majority of genes appears to be down-regulated ( $\log_2$  fold change  $<0$ ). In addition, the accumulation of points at low intensities may indicate additional technical problems, e.g., high background or low fluorescence intensities on the arrays.

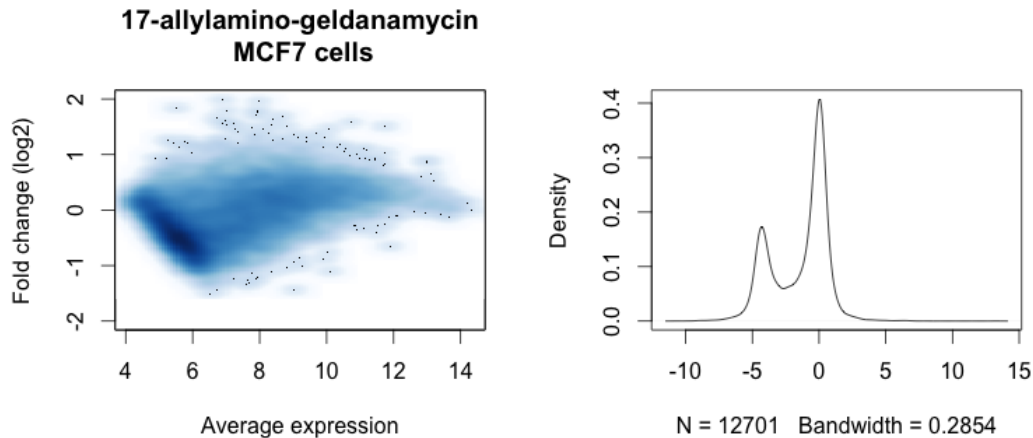


**Figure 21:** Distribution of z-scores for all assayed genes after Monorden treatment of MCF7 (left) and SKMEL5 cells (right).

The z-score distribution for this experiment is bi-modal, with the tallest peak shifted to the right of 0 by >2 units (figure 22, right) and a median-absolute deviation >1.4, due to a large numbers of genes with strongly negative z-scores. We typically remove instances whose z-score distribution mode is >0.8 units away from zero and/or displays a median absolute deviation >1.2. (These thresholds have been determined empirically and may need to be adjusted for other reference databases.)

```
> par(mfrow=c(1,2))
> smoothScatter(assayDataElement( GEOD5258.cmap, "exprs")["Exp26"],
+               assayDataElement( GEOD5258.cmap, "log_fc")["Exp26"],
+               xlab="Average expression", main="",
+               ylab="Fold change (log2)",
+               ylim=c(-2,2))
> z.distribution <- density( assayDataElement( GEOD5258.cmap, "z")["Exp26"])
> plot( z.distribution, main="")
> par(mfrow=c(1,1))
> title( "17-allylamino-geldanamycin\n MCF7 cells")

z.center <- z.distribution$x[which.max(z.distribution$y)]
z.center
[1] 2.145028
z.mad <- mad( assayDataElement( GEOD5258.cmap, "z")["Exp26"])
z.mad
[1] 1.422537
```



**Figure 22:** MA plot (left) and z-score distribution (right) showing the gene expression changes observed in MCF7 cells treated with allylamino-geldanamycin.

## References

- [1] Fisher, R. A. (1922). On the interpretation of  $\chi^2$  from contingency tables, and the calculation of  $p$ . *Journal of the Royal Statistical Society*, **85**(1), 87.
- [2] Goeman, J. J. and Bühlmann, P. (2007). Analyzing gene expression data in terms of gene sets: methodological issues. *Bioinformatics*, **23**(8), 980–7.
- [3] Jiang, Z. and Gentleman, R. (2007). Extensions to gene set enrichment. *Bioinformatics*, **23**(3), 306–13.
- [4] Kawata, K., Yokoo, H., Shimazaki, R., and Okabe, S. (2007). Classification of heavy-metal toxicity by human dna microarray analysis. *Environ Sci Technol*, **41**(10), 3769–74.
- [5] Lamb, J., Crawford, E. D., Peck, D., Modell, J. W., Blat, I. C., Wrobel, M. J., Lerner, J., Brunet, J.-P., Subramanian, A., Ross, K. N., Reich, M., Hieronymus, H., Wei, G., Armstrong, S. A., Haggarty, S. J., Clemons, P. A., Wei, R., Carr, S. A., Lander, E. S., and Golub, T. R. (2006). The connectivity map: using gene-expression signatures to connect small molecules, genes, and disease. *Science*, **313**(5795), 1929–35.
- [6] Lovén, J., Orlando, D. A., Sigova, A. A., Lin, C. Y., Rahl, P. B., Burge, C. B., Levens, D. L., Lee, T. I., and Young, R. A. (2012). Revisiting global gene expression analysis. *Cell*, **151**(3), 476–82.
- [7] Nakajima, S., Kato, H., Takahashi, S., Johno, H., and Kitamura, M. (2011). Inhibition of nf- $\kappa$ b by mg132 through er stress-mediated induction of lap and lip. *FEBS Lett*, **585**(14), 2249–54.
- [8] Nguyen, P. M., Park, M. S., Chow, M., Chang, J. H., Wrishnik, L., and Chan, W. K. (2010). Benzo[a]pyrene increases the nrf2 content by downregulating the keap1 message. *Toxicol Sci*, **116**(2), 549–61.
- [9] Sethi, G., Ahn, K. S., Pandey, M. K., and Aggarwal, B. B. (2007). Celestrol, a novel triterpene, potentiates tnf-induced apoptosis and suppresses invasion of tumor cells by inhibiting nf-kappab-regulated gene products and tak1-mediated nf-kappab activation. *Blood*, **109**(7), 2727–35.
- [10] Straus, D. S., Pascual, G., Li, M., Welch, J. S., Ricote, M., Hsiang, C. H., Sengchanthalangsy, L. L., Ghosh, G., and Glass, C. K. (2000). 15-deoxy-delta 12,14-prostaglandin j2 inhibits multiple steps in the nf-kappa b signaling pathway. *Proc Natl Acad Sci U S A*, **97**(9), 4844–9.
- [11] van Delft, J., Gaj, S., Lienhard, M., Albrecht, M. W., Kirpiy, A., Brauers, K., Claessen, S., Lizarraga, D., Lehrach, H., Herwig, R., and Kleinjans, J. (2012). Rna-seq provides new insights in the transcriptome responses induced by the carcinogen benzo[a]pyrene. *Toxicol Sci*, **130**(2), 427–39.