Vignette for *Mulder2012*: Diverse epigenetic strategies interact to control epidermal differentiation.

Xin Wang, Mauro A. Castro,
Klaas W. Mulder and Florian Markowetz

May 1, 2018

Contents

1 Introduction

The vignette helps the user to reproduce main results and figures of two applications of *PANs* (Posterior Association Networks) in [7]. In the first application, we predict a network of functional interactions between chromatin factors regulating epidermal stem cells. In the second application, we identify a gene module including many confirmed and highly promising therapeutic targets in Ewing's sarcoma. Please refer to bioconductor package *PANR* for detailed introduction about the methods, on which this package is dependent. Please also refer to Mulder et al. [5] and Arora et al. [4] for more details about the biological background, experimental design and data processing of the first and second applications, respectively.

2 Package installation

Please run all analyses in this vignette under version 2.14 of R. Prior to installation of package *Mulder2012*, R packages *HTSanalyzeR*, *org.Hs.eg.db*, *KEGG.db* (for enrichment analyses), *pvclust* (for hierarchical clustering with bootstrap resampling), *RedeR* (for network visualization) and *PANR* (the method package for inferring a posterior association network) should be installed. These packages can be installed directly from bioconductor:

```
> source("http://www.bioconductor.org/biocLite.R")
> biocLite(c("PANR", "RedeR", "pvclust", "HTSanalyzeR",
+ "org.Hs.eg.db", "KEGG.db"))
```

The 'snow' package is recommended to be installed for module searching by the *PANR* package more efficiently.

3 Application I–Step-by-step analysis

Mulder K. et al. studied the functions of known and predicted chromatin factors in self-renewal and differentiation of human epidermal stem cells [5]. To predict interactions between these chromatin factors, we propose posterior association networks (PANs) encoding gene functions on vertices and their functional associations on edges. Here we introduce step by step the computational workflow to perform mixture modelling, infer the functional

network, search for enriched gene modules, etc. starting from RNA interference screening data under multiple conditions.

3.1 Data preprocessing

In total, RNAi-based gene silencing was performed on 332 chromatin factors under five different conditions: vehicle, AG1478, BMP2/7, AG1478_BMP2/7 and serum stimulation. To quantify each gene's function in epidermal self-renewal, we measured the endogenous levels of transglutaminase I (TG1) protein, which is the key enzyme that mediates the assembly of the epidermal cornified envelope. First, we load the raw screening data:

```
> library(Mulder2012)
> library(HTSanalyzeR)
> library(PANR)

> data(Mulder2012.rawScreenData, package="Mulder2012")
> dim(rawScreenData)

[1] 680  18

> colnames(rawScreenData)

 [1] "PLATE"           "WELL"            "CHANNEL"
 [4] "vehicle_1"       "vehicle_2"       "vehicle_3"
 [7] "AG1478_1"        "AG1478_2"        "AG1478_3"
[10] "BMP2/7_1"        "BMP2/7_2"        "BMP2/7_3"
[13] "AG1478+BMP2/7_1" "AG1478+BMP2/7_2" "AG1478+BMP2/7_3"
[16] "serum10%_1"      "serum10%_2"      "serum10%_3"
```

To correct for plate-to-plate variability, the raw screening measurement x_{ki}^{TG1} for k-th well in plate i was normalized to DRAQ5 signal x_{ki}^{DRAQ5}, which was used as control, within the plate:

$$x_{ki}^{'} = \frac{x_{ki}^{TG1} - \overline{x}_i^{siTG1}}{x_{ki}^{DRAQ5}} \quad , \tag{1}$$

where $\overline{x}_i^{siTG1}$ denotes the mean of control signals in plate i. Z-scores were subsequently computed from the normalized data:

$$z_{ki} = \frac{x_{ki}^{'} - \mu_i}{\delta_i} \tag{2}$$

where μ_i and δ_i are the mean and standard deviation of all measurements within the ith plate.

This procedure is realised by function *Mulder2012.RNAiPre*:

```
> data(Mulder2012.rawScreenAnnotation, package="Mulder2012")
> Mulder2012<-Mulder2012.RNAiPre(rawScreenData, rawScreenAnnotation)
```

After the above preprocessing steps, we obtained a 332 (genes) $\times$ 15 (3 replicates $\times$ 5 conditions) matrix of z-scores.

```
> dim(Mulder2012)
```

```
[1] 332  15
```

```
> colnames(Mulder2012)
```

```
 [1] "vehicle"       "vehicle"       "vehicle"       "AG1478"
 [5] "AG1478"        "AG1478"        "BMP2/7"        "BMP2/7"
 [9] "BMP2/7"        "AG1478+BMP2/7" "AG1478+BMP2/7" "AG1478+BMP2/7"
[13] "serum10%"      "serum10%"      "serum10%"
```

3.2 Beta-mixture modelling

Modelling lack of association From the z-score matrix, we compute association scores between genes using uncentered correlation coefficients (also known as *cosine similarities*). The proposed Beta-mixture model is applied to quantify the significances of these associations. We first permuted the z-score matrix for 100 times, for each of which we compute cosine similarities and fit a null distribution to the density by maximum likelihood estimation using the function *fitdistr* in the R package *MASS* (Figure 1) [1]. The median values of the 100 fitted parameters were selected for modelling the ($\times$) component representing lack of association of the mixture distribution.

The fitting is done by the following codes:

```
> bm_Mulder2012<-new("BetaMixture", pheno=Mulder2012,
+ metric="cosine", order=1, model="global")
> bm_Mulder2012<-fitNULL(bm_Mulder2012, nPerm=100,
+ thetaNULL=c(alphaNULL=4, betaNULL=4), sumMethod="median",
+ permMethod="keepRep", verbose=TRUE)
```

To inspect the fitting performance, we can compare the density plots of the permuted screening data and fitted beta distribution:

```
> view(bm_Mulder2012, what="fitNULL")
```

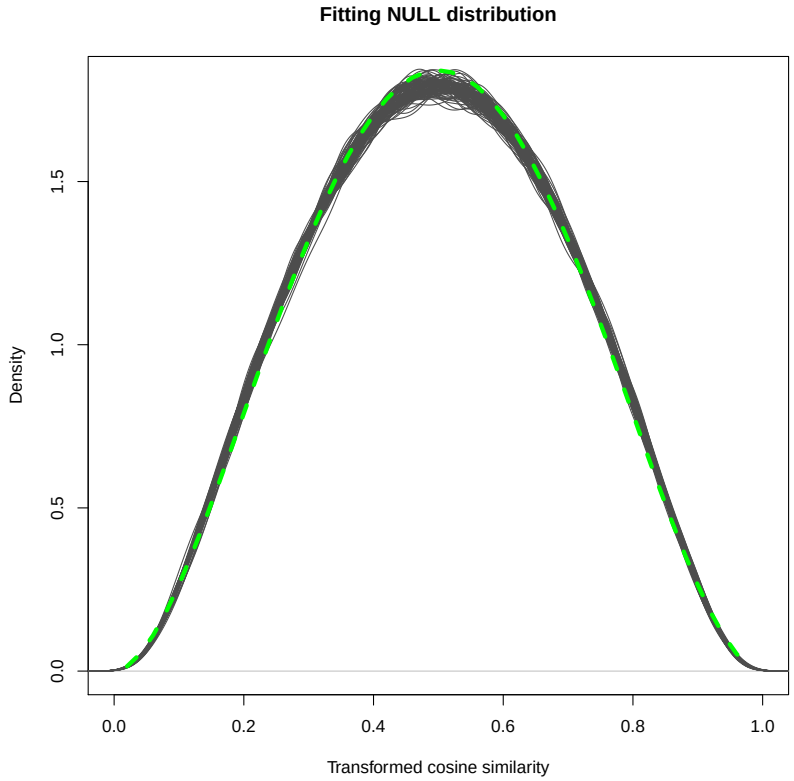

Figure 1: **Fit a beta distribution to association densities derived from permutations.** The screening data matrix is permuated for 100 times, and for each permuted data association densities were computed and a beta distribution was fitted. Each fitting result is plotted as a gray curve. The median scores of the two shape parameters estimated from permutations were selected to fix the $\times$ component (blue dashed curve).

Fitting a beta-mixture model to the real RNAi screens Having fixed the parameters for the component representing lack of associations, we do MLE to estimate the other parameters of the mixture model using EM algorithm. This step is conducted by the function *fitBM*:

```
> bm_Mulder2012<-fitBM(bm_Mulder2012, para=list(zInit=NULL,
+ thetaInit=c(alphaNeg=2, betaNeg=4,
+ alphaNULL=bm_Mulder2012@result$fitNULL$thetaNULL[["alphaNULL"]],
+ betaNULL=bm_Mulder2012@result$fitNULL$thetaNULL[["betaNULL"]],
+ alphaPos=4, betaPos=2), gamma=NULL),
+ ctrl=list(fitNULL=FALSE, tol=1e-3), verbose=TRUE)
```

Similarly, we can inspect the mixture modelling results (shown in Figure 2):

```
> view(bm_Mulder2012, what="fitBM")
```

Comparing the original histogram of cosine similarities, the fitted three beta distributions and the mixture of them, we found that the density of cosine similarities is successfully partitioned to three components capturing the population of noise (lack of association) and signal (positive or negative association). The posterior probabilies for each association belonging to different populations in the mixture model were computed subsequently for inference of the functional network.

3.3 Enrichment analyses

We hypothesize that genes interacting at the protein level may tend to have higher functional interaction. To test the hypothesis in this application, we conducted GSEA (Gene Set Enrichment Analysis) for protein-protein interactions (using PINdb [8]) in the posterior probabilities of associations following the three different components of the beta-mixture model. Not surprisingly, we observe highly significant enrichment of PPIs in the + and - components (p-values are 0.0067 and 0.0004, respectively) but not in the $\times$ component (p-value=1.0000) (Figure 3). Thus, protein-protein interactions can be potentially incorporated as *a priori* belief to achieve a better performance.

To run the enrichment analyses:

```
> PPI<-Mulder2012.PPIPre()
> Mulder2012.PPIenrich(pheno=Mulder2012, PPI=PPI$PPI,
+ bm=bm_Mulder2012)
```

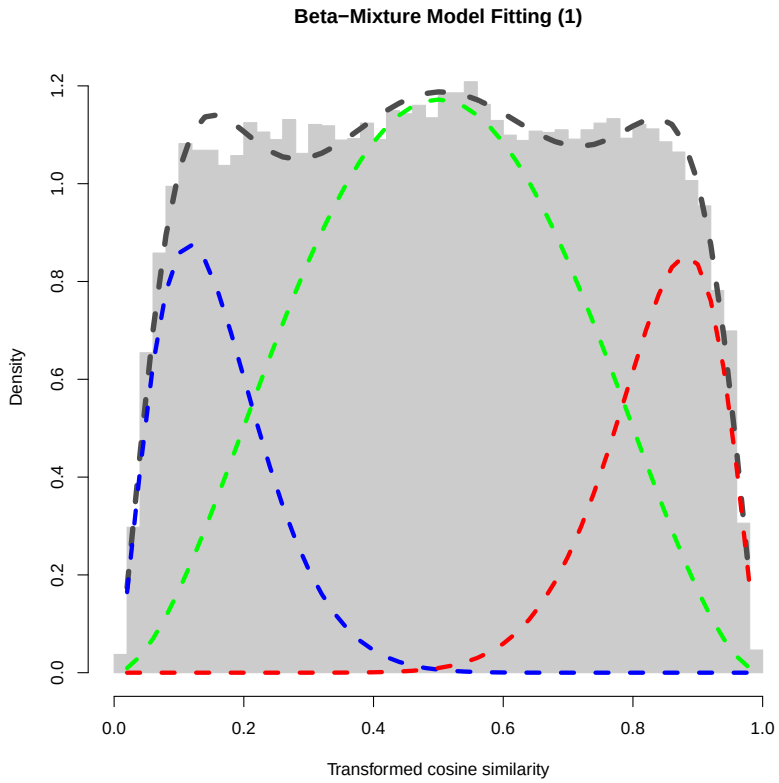

Figure 2: **Fit a beta-mixture model to association scores computed from the real screening data.** The fitting is conducted based on the EM algorithm with the shape parameters of the $\times$ component fixed by fitting to permuted screening data. The histogram and the dashed curves show the real distribution of transformed association scores and the fitting result, respectively. Fitted densities for positive, negative and nonexistent associations are illustrated by red, blue and green dashed curves, respectively.

The enrichment results can be view by:

```
> labels<-c("A", "B", "C")
> names(labels)<-c("neg", "none", "pos")
> for(i in c("neg", "none", "pos")) {
+ pdf(file=file.path("rslt", paste("fig5", labels[i], ".pdf",sep="")),
+ width=8, height=6)
+ GSEARandomWalkFig(Mulder2012, PPI, bm_Mulder2012, i)
+ graphics.off()
+ }
```

3.4 Incorporating protein-protein interactions

Here we take the advantage of prior knowledge such as protein-protein interactions to better predict functional connections using the extended model. Similarly, we first fit a null beta distribution to each of 100 permuted data sets, and used the median values of the fitted parameters to fix the $\times$ component in the mixture model. According to protein-protein interactions obtained from the PINdb database, we stratified the whole set of gene pairs to PPI group and non-PPI group. During the fitting to the extended model using the EM algorithm, different prior probabilities (mixture coefficients) for the three mixture components were used for these two groups.

```
> bm_ext<-new("BetaMixture", pheno=Mulder2012,
+ metric="cosine", order=1, model="stratified")
> bm_ext<-fitNULL(bm_ext, nPerm=100,
+ thetaNULL=c(alphaNULL=4, betaNULL=4),
+ sumMethod="median", permMethod="keepRep", verbose=TRUE)
> bm_ext<-fitBM(bm_ext, para=list(zInit=NULL,
+ thetaInit=c(alphaNeg=2, betaNeg=4,
+ alphaNULL=bm@result$fitNULL$thetaNULL[["alphaNULL"]],
+ betaNULL=bm@result$fitNULL$thetaNULL[["betaNULL"]],
+ alphaPos=4, betaPos=2), gamma=NULL),
+ ctrl=list(fitNULL=FALSE, tol=1e-3), verbose=TRUE)
```

As expected, the fitted mixture coefficients of the + and − components for the PPI group (30.4% and 30.9%) are significantly higher than the non-PPI group (18.2% and 17.9%) (Figure 4). The fitting results suggest that gene

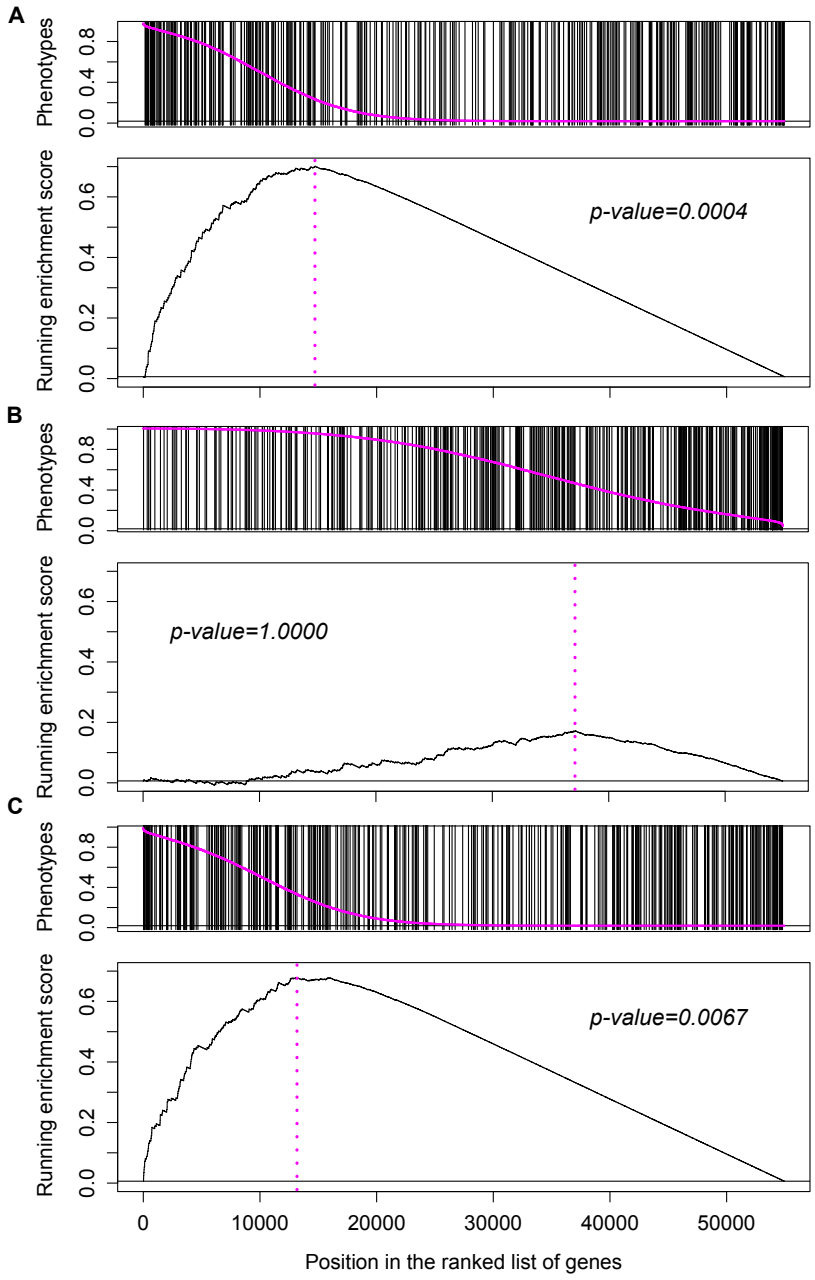

Figure 3: **Enrichment of protein-protein interactions in posterior probabilities.** (A), (B) and (C) correspond to the enrichment of protein-protein interactions (PPIs) in the posterior probabilities for associations belonging to the +, - and × component, respectively. In each one of the three figures, the upper panel shows the ranked phenotypes by a pink curve and the positions of PPIs in the ranked phenotypes, while the lower panel shows the running sum scores of GSEA (Gene Set Enrichment Analysis) random walks [3].

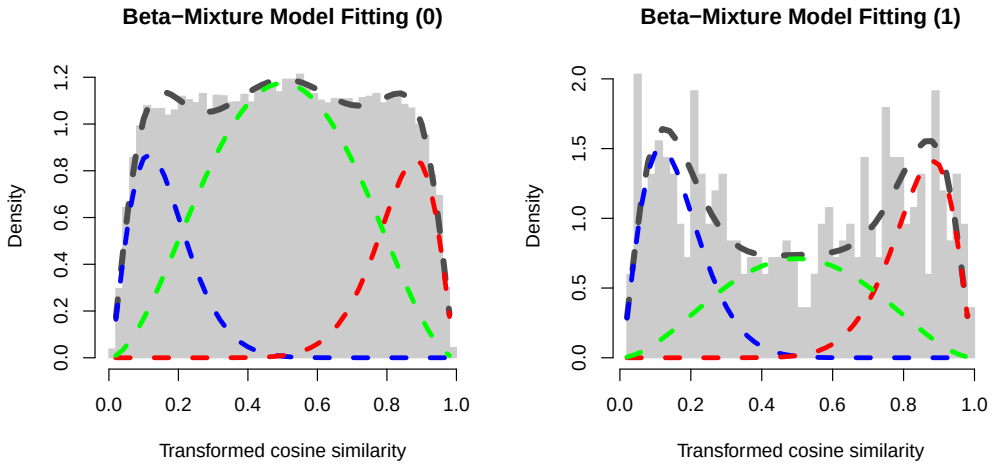

Figure 4: **Fitting results of the extended beta-mixture model.** The whole set of gene pairs are stratified to PPI(protein-protein interaction) group and non-PPI group. The extended beta-mixture model is fitted to functional associations, setting different prior probabilities (mixture coefficients) to these two groups. The fitting results for the PPI group is illustrated in (A), and the non-PPI group in (B). The histogram and the dashed curves show the real distribution of transformed association scores and the fitting result, respectively. Fitted distributions for positive, negative and lack of association are illustrated by red, blue and green dashed curves, respectively. The fitting results suggest that gene pairs in the PPI group have higher probability to be functionally connected than the non-PPI group.

pairs in the PPI group are much more likely to be positively or negatively associated.

```
> view(bm_ext, "fitBM")
```

3.5 Inferring a posterior association network

To infer a posterior association network, we compute signal-to-noise ratios (SNR), which are posterior odds of gene pairs in favor of signal (association) to noise (lack of association). Setting the cutoff score for SNR at 10, we

filtered out non-significant gene associations and obtain a very sparse functional network (Figure 5). This procedure is accomplished by the following codes:

```
> pan_ext<-new("PAN", bm1=bm_ext)
> pan_ext<-infer(pan_ext, para=
+ list(type="SNR", log=TRUE, sign=TRUE,
+ cutoff=log(10)), filter=FALSE, verbose=TRUE)
```

PAN provides a function *buildPAN* to build an *igraph* object for visualization:

```
> pan_ext<-buildPAN(pan_ext, engine="RedeR",
+ para=list(nodeSumCols=1:3, nodeSumMethod="average",
+ hideNeg=TRUE))
```

To view the predicted network in *RedeR*, we can use the function *viewPAN*:

```
> library(RedeR)
> viewPAN(pan_ext, what="graph")
```

As shown in Figure 5, the network is naturally splitted to two clusters consisting of genes with positive and negative perturbation effects, respectively.

3.6 Searching for enriched functional modules

We next conduct second-order hierarchical clustering to search for enriched functional modules. To assess the uncertainty of the clustering analysis, we perform multiscale bootstrap resampling (more details in the R package *pvclust* [2]).

To make it more efficient, we recommend to use the 'snow' package for parallel computing:

```
> library(snow)
> ##initiate a cluster
> options(cluster=makeCluster(4, "SOCK"))
```

Please note that to enable second-order clustering in package *pvclust*, we have to modify function *dist.pvclust* and *parPvclust* using the following code:

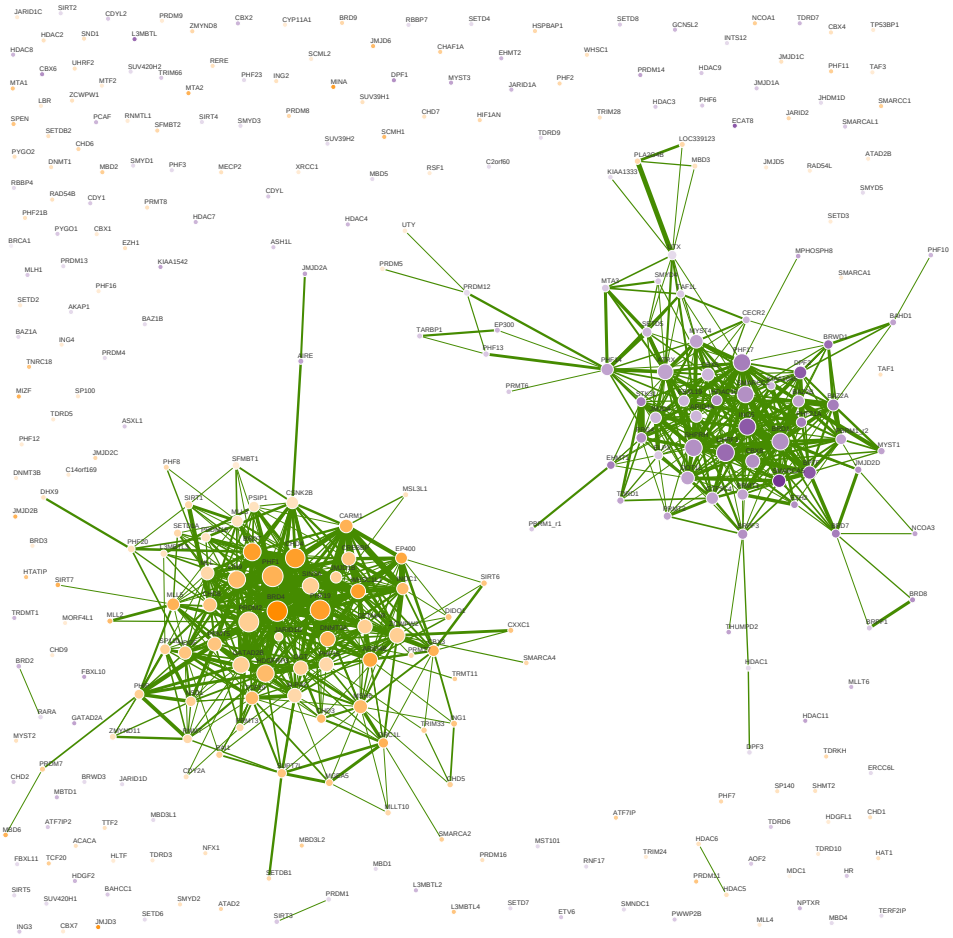

Figure 5: **Predicted association network of functional interactions.** This figure presents the predicted significant possitive functional interactions between 158 chromatin factors (SNR>10). Nodes with purple and orange colors represent positive and negative perturbation effects, respectively. Node colors are scaled according to their averaged perturbation effects under the vehicle condition. Node sizes are scaled according to their degrees. Edge widths are in proportion to log signal-to-noise ratios. Edges are colored in green representing positive associations between genes.

```
> ns<-getNamespace("pvclust")
> en<-as.environment("package:pvclust")
> assignInNamespace("dist.pvclust", dist.pvclust4PAN,
+ ns="pvclust", envir=ns)
> dist.pvclust<-getFromNamespace("dist.pvclust",
+ ns=getNamespace("pvclust"))
> unlockBinding("parPvclust", ns)
> assignInNamespace("parPvclust", parPvclust4PAN,
+ ns="pvclust", envir=ns)
> lockBinding("parPvclust", ns)
> parPvclust<-getFromNamespace("parPvclust", ns)
> if(is(getOption("cluster"), "cluster") &&
+ "package:snow" %in% search()) {
+ clusterCall(getOption("cluster"), assignInNamespace,
+ x="dist.pvclust", value=dist.pvclust4PAN, ns=ns)
+ clusterCall(getOption("cluster"), unlockBinding,
+ sym="parPvclust", env=ns)
+ clusterCall(getOption("cluster"), assignInNamespace,
+ x="parPvclust", value=parPvclust4PAN, ns=ns)
+ clusterCall(getOption("cluster"), lockBinding,
+ sym="parPvclust", env=ns)
+ }

> pan_ext<-pvclustModule(pan=pan_ext, nboot=10000,
+ metric="cosine2", hclustMethod="average", filter=FALSE)
> ##stop the cluster
> stopCluster(getOption("cluster"))
> options(cluster=NULL)
```

In the code above, please ensure that the name of the cluster is 'cluster', as it will be recognized inside the 'PANR' package.

With 10,000 times' resampling we obtained 39 significant clusters (p-value < 0.05) including more than three genes. Mapping these gene clusters to the inferred functional network, we identified 13 tightly connected modules (density > 0.5). To view these modules in a compact way (Figure 6), the user can use the following function:

```
> rdp <- RedPort('MyPort')
> calld(rdp)
```

```
> Mulder2012.module.visualize(rdp, pan_ext, mod.pval.cutoff=0.05,
+ mod.size.cutoff=4, avg.degree.cutoff=0.5)
```

Among these modules the one cosisting of *ING5*, *UHRF1*, *EZH2*, *SMARCA5*, *BPTF*, *SMARCC2* and *PRMT1* is of particular interest, as it indicates a functional connection between *BPTF*, *EZH2*, NURF and MORF complexes, which have been independently implicated in epidermal self-renewal. Further combinatorial knock-down experiments validated many genetic interactions between *ING5*, *BPTF*, *SMARCA5*, *EZH2* and *UHRF1*. These biological validations demonstrate the power of the proposed integrative computational approach for predicting association networks of functional gene interactions and searching for enriched gene modules.

4 Application I–pipeline functions

We provide two pipeline functions for reproducing all the data and figures. Please note that to enable second-order hierarchical clustering in *pvclust*, function *dist.pvclust* and *parPvclust* must be modified using the code described in section 3.6.

4.1 Pipeline function to reproduce data

All data can be recomputed by a pipeline function *Mulder2011.pipeline*:

```
> Mulder2012.pipeline(
+ par4BM=list(model="global", metric="cosine", nPerm=20),
+ par4PAN=list(type="SNR", log=TRUE, sign=TRUE,
+ cutoff=log(10), filter=FALSE),
+ par4ModuleSearch=list(nboot=10000, metric="cosine2",
+ hclustMethod="average", filter=FALSE)
+ )
```

This pipeline includes the following analysis procedures:

- Data pre-processing including computing z-scores from the raw RNAi screening data and extracting protein-protein interaction information from the PINdb database [8] (details in section 3.1).

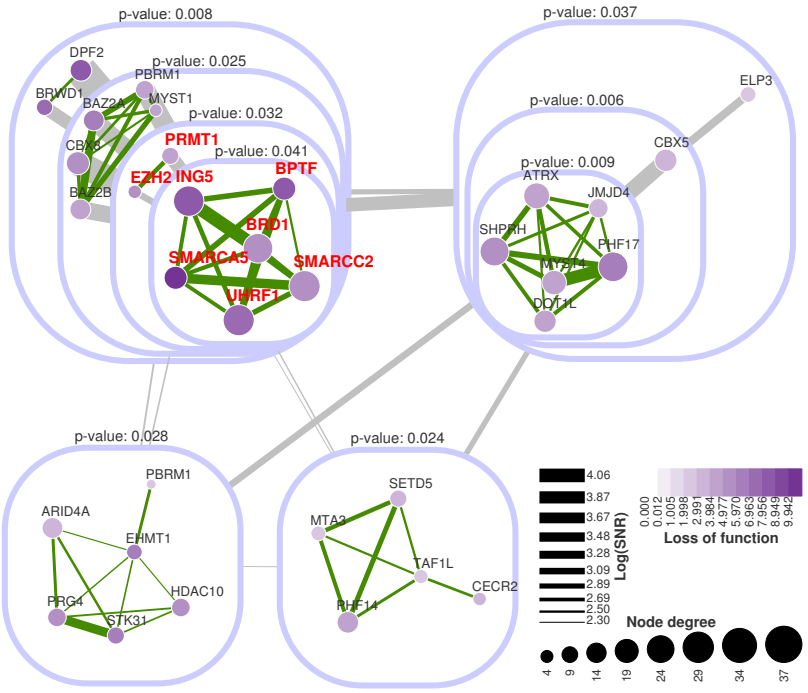

Figure 6: **Top significant modules predicted by PANs .** Nodes with purple colors represent positive perturbation effects. Node colors are scaled according to their averaged perturbation effects under the vehicle condition. Node sizes are scaled in proportion to their degrees. Edge widths are in proportion to log signal-to-noise ratios. Edges colored in green and grey represent positive interactions inside modules and summed interactions between modules, respectively. This figure illustrates top significant modules and their dense functional interactions. Genes colored in red were selected for further experimental investigation.

- Fitting a beta distribution to association densities computed from permuted screening data. The fitted shape parameters will be used to model the component representing lack of association in the beta-mixture model (details in section 3.2).
- Fitting a beta-mixture model to association scores computed from the real screening data (details in section 3.2).
- Enrichment analyses of posterior probabilities of gene pairs belonging to a positive, negative or lack of association in protein-protein interactions in the nucleus (details in section 3.3).
- Fitting a stratified beta-mixture model to incorporate protein-protein interactions (details in section 3.4).
- Inferring an association network of functional interactions between chromatin factors based on the beta-mixture modelling results (details in section 3.5).
- Searching for significant gene modules based on hierarchical clustering with multiscale resampling (details in section 3.6).

4.2 Pipeline function to reproduce figures

All figures can be regenerated, some of which need manual improvement, using the following function:

```
> Mulder2012.fig(what="ALL")
```

in which `what` specifies which figure to generate:

- 'NULLfitting' (Fig. 4A in [7]): density curves of transformed cosine similarities computed from permuted screening data and fitted beta distribution. This figure can be used to assess the fitting of the $\times$ component in the beta-mixture model.
- 'BMfitting' (Fig. 4B in [7]): a histogram of transformed cosine similarities computed from the real screening data, fitted beta-mixture distribution as well as mixture coefficient weighted beta distributions fitted for the three components. This figure is also used for assessing the fitting of the beta-mixture model to screening data.

- 'PPIenrich' (Fig. 5 in [7]): figures of the enrichment analyses results. Each figure shows the positions of the protein-protein interactions in the ranked phenotypes (posterior probabilities) and the running enrichment scores.
- 'sigMod' (Fig. 7 in [7]): a figure of top significant gene modules searched by multiscale bootstrap resampling analyses using *pvclust*.
- 'selMod' (Fig. 8A in [7]): a figure of the module selected for further validation using combinatorial knock-down experiments.

5 Application II–Step-by-step analysis

In this application, we use RNAi phenotyping screens across multiple cell lines to infer functional modules of kinases that are critical for growth and proliferation of Ewing's sarcoma. We demonstrate that our model can make efficient use of single gene perturbation data to predict robust functional interactions. The data used in this case study is a matrix (572×8) of Z-scores from high throughput RNAi screens run in duplicates targeting 572 human kinases in four Ewing's sarcoma cell lines: TC-32, TC-71, SK-ES-1 and RD-ES [4]. In these phenotyping screens, viability was assessed using a luminescence-based cell to quantify each gene's function in cancer cell growth and proliferation. The screening data was corrected for plate row variations and normalized using Z-score method as described in [4]. Compared to other RNAi screens in normal human fibroblast cell line, the four Ewing's sarcoma cell lines exhibited significant similarities, suggesting robust and consistent functional interactions among perturbed genes across cell lines [4].

We first load the screening data:

```
> data(Arora2010, package="Mulder2012")
> dim(Arora2010)

[1] 572   8

> colnames(Arora2010)

[1] "TC-32"   "TC-32"   "TC-71"   "TC-71"   "SK-ES-1" "SK-ES-1"
[7] "RD-ES"   "RD-ES"
```

5.1 Beta-mixture modelling

To predict the functional interactions between genes, a beta-mixture model was applied to quantify the significance of their associations, which are measured by cosine similarities computed from the Z-score matrix. We first permuted the Z-score matrix 20 times, computing cosine similarities and fitting a null distribution by maximum likelihood estimation (Figure 7). The median values of the 20 fitted parameters were selected to fix the $\times$ component representing lack of association in the mixture model.

```
> bm_Arora2010<-new("BetaMixture", pheno=Arora2010,
+ metric="cosine", order=1, model="global")
> bm_Arora2010<-fitNULL(bm_Arora2010, nPerm=20,
+ thetaNULL=c(alphaNULL=4, betaNULL=4), sumMethod="median",
+ permMethod="keepRep", verbose=TRUE)

> view(bm_Arora2010, what="fitNULL")
```

Having fixed the parameters for the $\times$ component, we performed MAP inference with an uninformative prior (uniform Dirichlet priors) to estimate the other parameters of the global mixture model using the EM algorithm.

```
> bm_Arora2010<-fitBM(bm_Arora2010, para=list(zInit=NULL,
+ thetaInit=c(alphaNeg=2, betaNeg=4,
+ alphaNULL=bm_Arora2010@result$fitNULL$thetaNULL[["alphaNULL"]],
+ betaNULL=bm_Arora2010@result$fitNULL$thetaNULL[["betaNULL"]],
+ alphaPos=4, betaPos=2), gamma=NULL),
+ ctrl=list(fitNULL=FALSE, tol=1e-3), verbose=TRUE)
```

Comparing the original histogram of cosine similarities, the fitted three beta distributions and the mixture of them (Figure 8), we found that the distribution of cosine similarities is successfully partitioned to three components capturing the population of signal (positive or negative association) and noise (lack of association).

```
> view(bm_Arora2010, what="fitBM")
```

The posterior probabilities for each association belonging to different populations in the mixture model were computed subsequently for inference of the functional network.

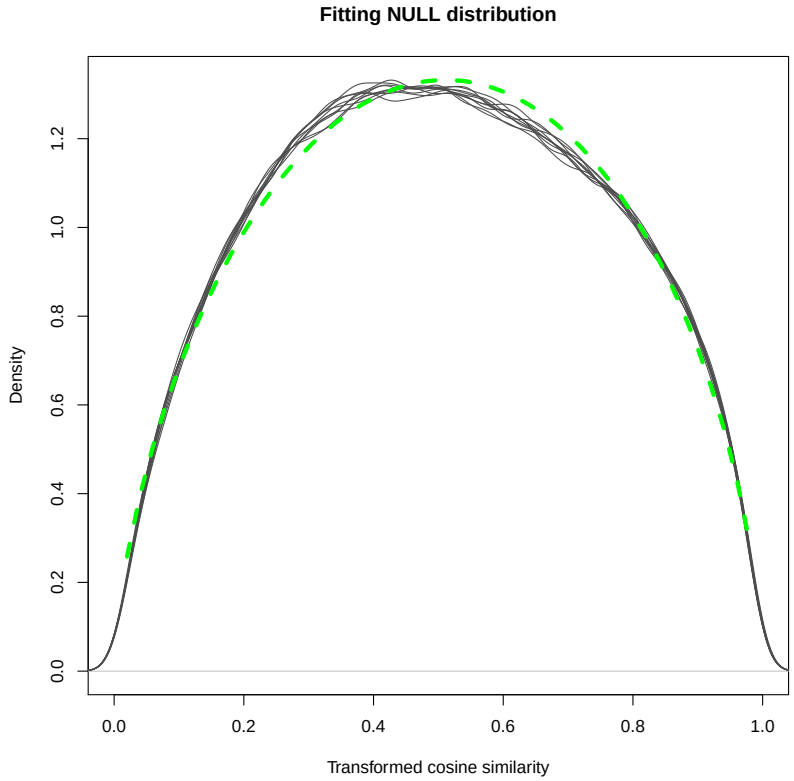

Figure 7: **Fit a beta distribution to association densities derived from permutations.** The screening data matrix is permuated for 20 times, and for each permuted data association densities were computed and a beta distribution was fitted. Each fitting result is plotted as a gray curve. The median scores of the two shape parameters estimated from permutations were selected to fix the $\times$ component (blue dashed curve).

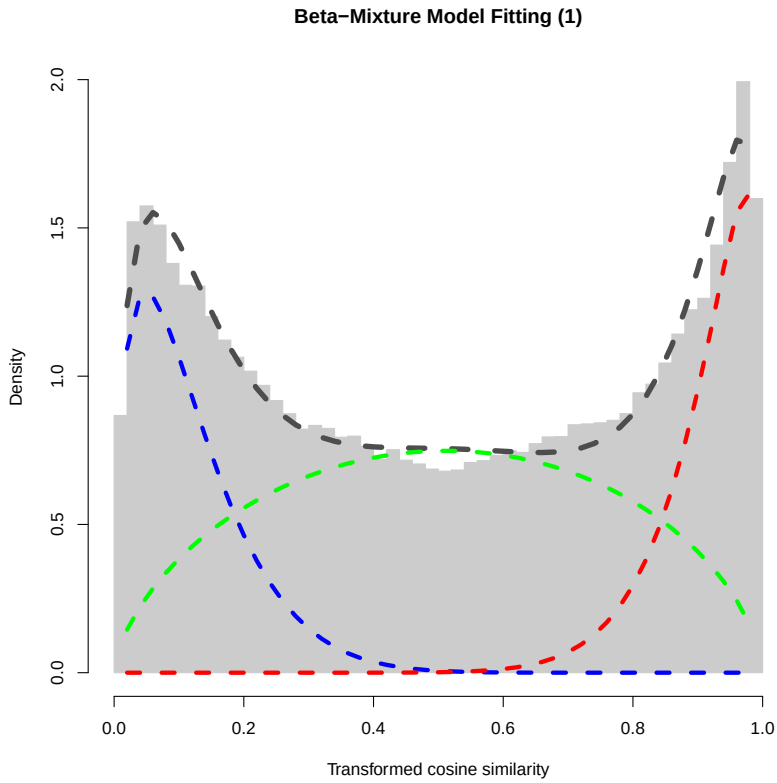

Figure 8: **Fit a beta-mixture model to association densities derived from the real screening data.** The fitting is conducted based on the EM algorithm with the shape parameters of the $\times$ component fixed by fitting to permuted screening data. The histogram and the dashed curves show the real distribution of transformed association scores and the fitting result, respectively. Fitted densities for positive, negative and nonexistent associations are illustrated by red, blue and green dashed curves, respectively.

5.2 Inferring a posterior association network

Having fitted the global mixture model to data successfully, we inferred a network of functional interactions between kinases based on the proposed edge inference approach. Setting the cutoff SNR score at 10, we filtered out non-significant edges and obtained a very sparse network. This procedure is accomplished by the following codes:

```
> pan_Arora2010<-new("PAN", bm1=bm_Arora2010)
> pan_Arora2010<-infer(pan_Arora2010, para=
+ list(type="SNR", log=TRUE, sign=TRUE,
+ cutoff=log(10)), filter=FALSE, verbose=TRUE)
```

PAN provides a function *buildPAN* to build an *igraph* object for visualization:

```
> pan_Arora2010<-buildPAN(pan_Arora2010, engine="RedeR",
+ para=list(nodeSumCols=1:2, nodeSumMethod="average",
+ hideNeg=TRUE))
```

To view the predicted network in *RedeR*, we can use the function *viewPAN*:

```
> viewPAN(pan_Arora2010, what="graph")
```

As shown in Figure 9, the network is naturally splitted to two clusters consisting of genes with positive and negative perturbation effects, respectively.

5.3 Searching for enriched functional modules

Hierarchical clustering with multiscale bootstrap resampling was conducted subsequently using the R package *pvclust* [2]. With 10000 times' resampling, we obtained 65 significant (p-value < 0.05) clusters with more than four genes. Of all these significant clusters, 30 clusters are enriched for functional interactions (module density > 0.5).

```
> library(snow)
> ##initiate a cluster
> options(cluster=makeCluster(4, "SOCK"))
> pan_Arora2010<-pvclustModule(pan_Arora2010, nboot=10000,
```

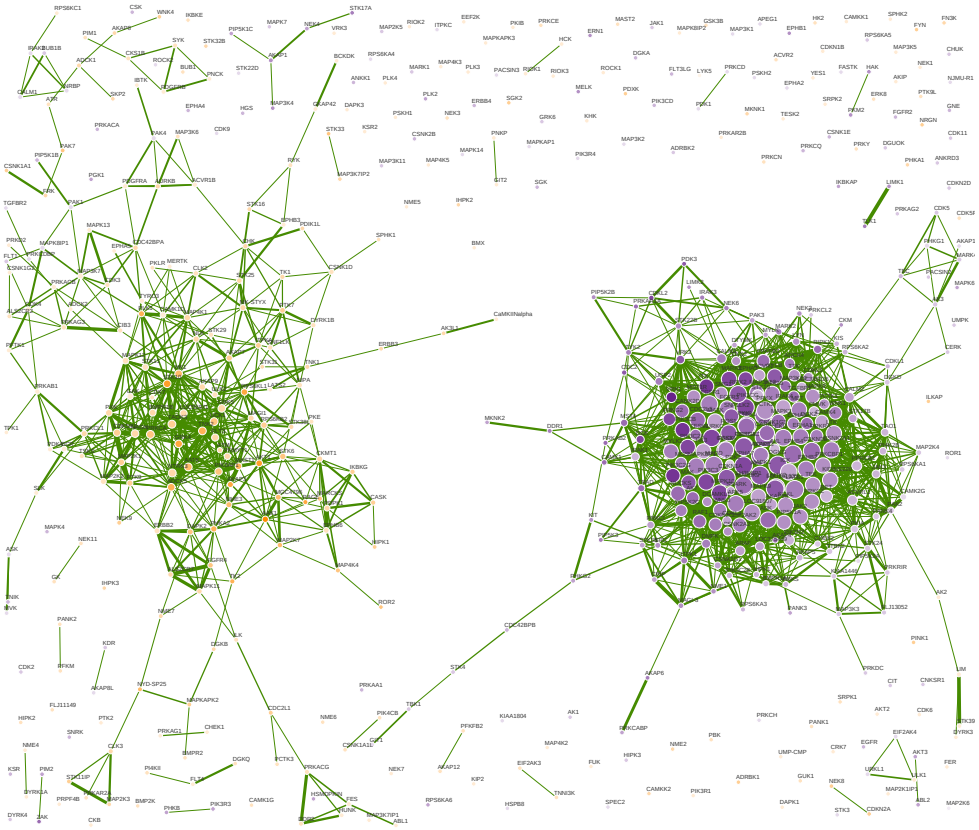

Figure 9: **Predicted association network of functional interactions.** This figure presents the predicted significant possitive functional interactions between 158 chromatin factors (SNR>10). Nodes with purple and orange colors represent positive and negative perturbation effects, respectively. Node colors are scaled according to their averaged perturbation effects under the vehicle condition. Node sizes are scaled according to their degrees. Edge widths are in proportion to log signal-to-noise ratios. Edges are colored in green representing positive associations between genes.

```
+ metric="consine2", hclustMethod="average", filter=TRUE,
+ verbose=TRUE, r=c(6:12/8))
> ##stop the cluster
> stopCluster(getOption("cluster"))
> options(cluster=NULL)
```

These clusters are superimposed to predicted posterior association networks to build functional modules. To visualize these modules in *RedeR* (Figure 10):

```
> rdp <- RedPort('MyPort')
> calld(rdp)
> Arora2010.module.visualize(rdp, pan_Arora2010, mod.pval.cutoff=
+ 0.05, mod.size.cutoff=4, avg.degree.cutoff=0.5)
```

5.4 Pathway analysis

Previous RNAi screening studies such as [4] were dedicated to discovering single genes that are pivotal for inhibiting Ewing's sarcoma. In our predictions, genes in the module are densely connected with highly siginificant functional interactions, indicating possible genetic interactions may exist among them. If the hypothesis is true, these genes may be involved in the same biological processes. Focusing on genes in the second module, we further searched for kinase pathways in which they are enriched. Hypergeometric tests were performed on all genes in this module to test their overrepresentation in KEGG pathways using R package *HTSanalyzeR* [6]. In total, we identified 15 significant KEGG pathways (Benjamini-Hochberg adjusted p-value < 0.05) with more than two observed hits (Figure 11).

```
> pw.Arora2010<-Arora2010.hypergeo(pan_Arora2010, mod.pval.cutoff=0.05,
+ mod.size.cutoff=4, avg.degree.cutoff=0.5)
> pw.Arora2010<-pw.Arora2010[[1]]
> obs.exp<-as.numeric(pw.Arora2010[, 4])
> names(obs.exp)<-paste(as.character(pw.Arora2010[, 6]), "  (",
+ format(pw.Arora2010[, 5], scientific=TRUE, digits=3), ")", sep="")
> par(mar=c(4, 16, 1, 1), cex=0.8)
> barplot((obs.exp), horiz=TRUE, las=2, xlab="Observed/Expected Hits",
+ cex.axis=0.8)
```

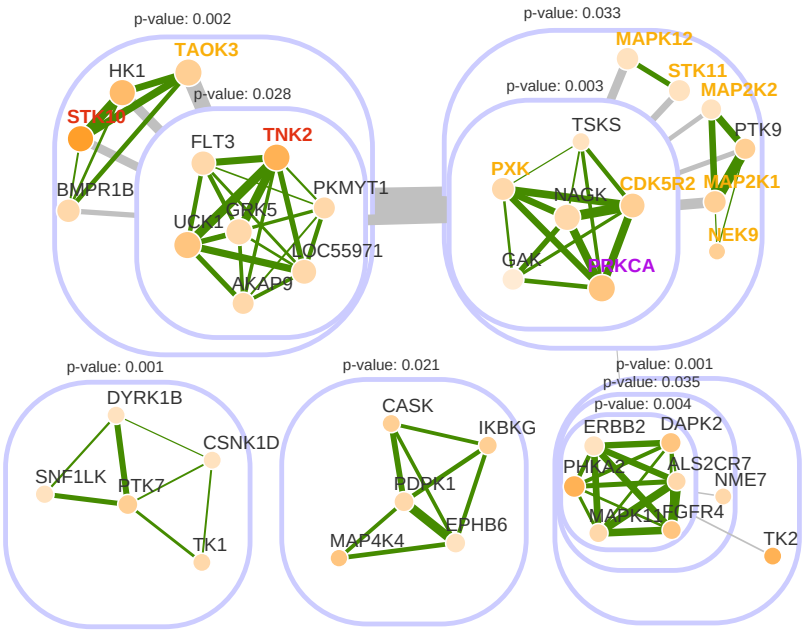

Figure 10: **Top significant modules predicted by PANs in Ewing's sarcoma.** The significant modules predicted by *PAN* are presented in a nested structure. Each module is illustrated by a rounded rectangle including sub-modules and/or individual genes. The p-value (on the top of each module) computed by *pvclust* indicates the stability of genes being clustered together. *PRCKA* (the gene colored in purple) is known to be a kinase target for human sarcomas, and an inhibitor PKC412 targeting *PRCKA* has already been tested in the clinic. *STK10* and *TNK2* (colored in red) in the upper left module have been identified as potential therapeutic targets. Another eight genes (colored in yellow) in the upper left and right modules were also highly associated with apoptosis of Ewing's sarcoma.

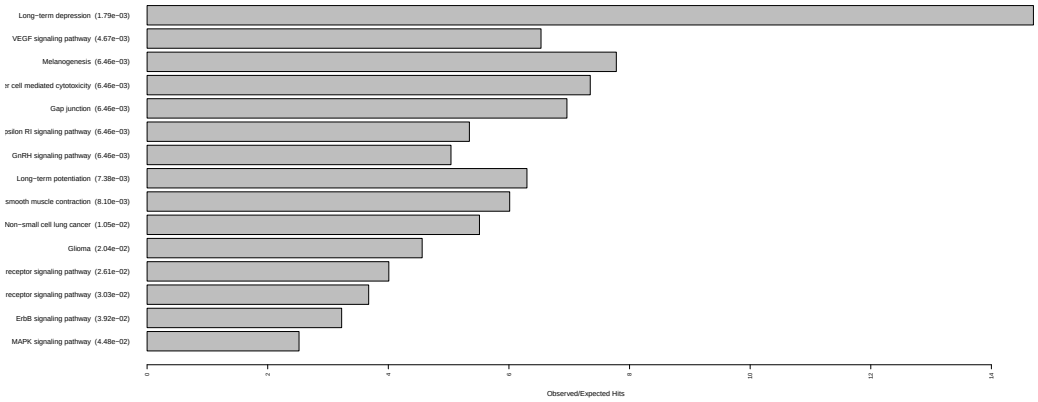

Figure 11: **Significantly overrepresented KEGG pathways.** Hypergeometric tests were performed to evaluate overrepresentation of genes included in the upper right module in human KEGG pathways. Top significant pathways (p-value < 0.05) are ranked by p-value increasingly, and their corresponding ratios of the number of observed hits to expected hits are illustrated by a bar plot.

6 Application II–pipeline functions

Similar to the first application, we provide two pipeline functions to reproduce all results and figures. Please note again that to enable second-order hierarchical clustering in *pvclust*, function *dist.pvclust* and *parPvclust* must be modified using the code described in section 3.6.

6.1 Pipeline function to reproduce data

All data can be recomputed by a pipeline function *Arora2010.pipeline*:

```
> Arora2010.pipeline(
+ par4BM=list(model="global", metric="cosine", nPerm=20),
+ par4PAN=list(type="SNR", log=TRUE, sign=TRUE,
+ cutoff=log(10), filter=FALSE),
+ par4ModuleSearch=list(nboot=10000, metric="cosine2",
+ hclustMethod="average", filter=FALSE)
+ )
```

This pipeline includes the following analysis procedures:

- Fitting a beta distribution to association scores computed from permuted screening data. The fitted shape parameters will be used to model the $\times$ component in the beta-mixture model (details in section 5.1).
- Fitting a beta-mixture model to association scores computed from the real screening data (details in section 5.1).
- Inferring a posterior association network for Ewing's sarcoma based on the beta-mixture modelling results (details in section 5.2).
- Searching for significant functional modules based on hierarchical clustering with multiscale resampling (details in section 5.3).
- Performing hypergeometric tests to evaluate overrepresentation of genes included in the module of interest in human KEGG pathways (details in section 5.4).

6.2 Pipeline function to reproduce figures

Most figures can be regenerated using the following function:

```
> Arora.fig(what="ALL")
```

in which `what` specifies which figure to generate:

- 'NULLfitting' (Fig. 3A in [7]): density curves of transformed cosine similarities computed from permuted screening data and fitted beta distribution. This figure can be used to assess the fitting of the $\times$ component in the beta-mixture model.
- 'BMfitting' (Fig. 3B in [7]): a histogram of transformed cosine similarities computed from the real screening data, fitted beta-mixture distribution as well as mixture coefficient weighted beta distributions fitted for the three components. This figure is also used for assessing the fitting of beta-mixture model to screening data.
- 'sigMod' (Fig. 3C in [7]): a figure of top significant gene modules identified by hierarchical clustering with multiscale bootstrap resampling using *pvclust*.

- 'pathway' (Fig. 3D in [7]): a figure illustrating significantly overrepresented KEGG pathways.

7 Session info

This document was produced using:

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

- R version 3.5.0 (2018-04-23), `x86_64-pc-linux-gnu`
- Locale: `LC_CTYPE=en_US.UTF-8`, `LC_NUMERIC=C`, `LC_TIME=en_US.UTF-8`, `LC_COLLATE=C`, `LC_MONETARY=en_US.UTF-8`, `LC_MESSAGES=en_US.UTF-8`, `LC_PAPER=en_US.UTF-8`, `LC_NAME=C`, `LC_ADDRESS=C`, `LC_TELEPHONE=C`, `LC_MEASUREMENT=en_US.UTF-8`, `LC_IDENTIFICATION=C`
- Running under: `Ubuntu 16.04.4 LTS`
- Matrix products: default
- BLAS: `/home/biocbuild/bbs-3.7-bioc/R/lib/libRblas.so`
- LAPACK: `/home/biocbuild/bbs-3.7-bioc/R/lib/libRlapack.so`
- Base packages: base, datasets, grDevices, graphics, methods, parallel, stats, stats4, utils
- Other packages: AnnotationDbi 1.42.0, Biobase 2.40.0, BiocGenerics 0.26.0, HTSanalyzeR 2.32.0, IRanges 2.14.0, KEGG.db 3.2.3, Mulder2012 0.20.0, PANR 1.26.0, S4Vectors 0.18.0, igraph 1.2.1, org.Hs.eg.db 3.6.0
- Loaded via a namespace (and not attached): BioNet 1.40.0, BiocInstaller 1.30.0, Category 2.46.0, DBI 0.8, DEoptimR 1.0-8, GSEABase 1.42.0, MASS 7.3-50, Matrix 1.2-14, R6 2.2.2, RBGL 1.56.0, RColorBrewer 1.1-2, RCurl 1.95-4.10, RSQLite 2.1.0, RankProd 3.6.0, Rcpp 0.12.16, RedeR 1.28.0, Rmpfr 0.7-0, XML 3.98-1.11, affy 1.58.0, affyio 1.50.0, annotate 1.58.0, assertthat 0.2.0, biomaRt 2.36.0, bit 1.1-12, bit64 0.9-7, bitops 1.0-6,

blob 1.1.1, cellHTS2 2.44.0, cluster 2.0.7-1, colorspace 1.3-2, compiler 3.5.0, digest 0.6.15, genefilter 1.62.0, ggplot2 2.2.1, gmp 0.5-13.1, graph 1.58.0, grid 3.5.0, gtable 0.2.0, httr 1.3.1, hwriter 1.3.2, lattice 0.20-35, lazyeval 0.2.1, limma 3.36.0, locfit 1.5-9.1, magrittr 1.5, memoise 1.1.0, munsell 0.4.3, mvtnorm 1.0-7, pcaPP 1.9-73, pillar 1.2.2, pkgconfig 2.0.1, plyr 1.8.4, prada 1.56.0, preprocessCore 1.42.0, prettyunits 1.0.2, progress 1.1.2, pvclust 2.0-0, rlang 0.2.0, robustbase 0.93-0, rrcov 1.4-3, scales 0.5.0, splines 3.5.0, splots 1.46.0, stringi 1.1.7, stringr 1.3.0, survival 2.42-3, tibble 1.4.2, tools 3.5.0, vsn 3.48.0, xtable 1.8-2, zlibbioc 1.26.0

8 References

[1] W. N. Venables and B. D. Ripley (2002). Modern Applied Statistics with S (Fourth edition). *Springer*, ISBN 0-387-95457-0. 5

[2] Suzuki, R. and Shimodaira, H. (2006). Pvclust: an R package for assessing the uncertainty in hierarchical clustering. *Bioinformatics*, **22**(12), 1540. 12, 22

[3] Subramanian, A. and Tamayo, P. et al. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *PNAS*, **102**(43), 15545. 10

[4] Arora, S. and Gonzales, I.M. and Hagelstrom, R.T. and Beaudry, C. and Choudhary, A. and Sima, C. and Tibes, R. and Mousses, S. and Azorsa, D.O. and others (2010). RNAi phenotype profiling of kinases identifies potential therapeutic targets in Ewing's sarcoma. *Molecular Cancer*, **9**(1), 218. 3, 18, 24

[5] Klaas W. Mulder, Xin Wang, Carles Escriu, Yoko Ito, Roland F. Schwarz, Jesse Gillis, Gabor Sirokmany, Giacomo Donati, Santiago Uribe-Lewis, Paul Pavlidis, Adele Murrell, Florian Markowetz and Fiona M. Watt

(2012). Diverse epigenetic strategies interact to control epidermal differentiation. *doi:10.1038/ncb2520*. 3

[6] Wang, X. and Terfve, C. and Rose, J.C. and Markowetz, F. (2011). HTSanalyzeR: an R/Bioconductor package for integrated network analysis of high-throughput screens. *Bioinformatics*, **27**(6), 879–880. 24

[7] Wang, X. and Castro, M.A. and Mulder, K. and Markowetz, F. (2012). Posterior association networks and functional modules inferred from rich phenotypes of gene perturbations. *doi:10.1371/journal.pcbi.1002566*. 3, 17, 18, 27, 28

[8] Luc PV and Tempst P (2004). PINdb: a database of nuclear protein complexes from human and yeast. *Bioinformatics*, **20**(9), 1413. 7, 15

[9] Eisen, M.B. and Spellman, P.T. and Brown, P.O. and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, **95**(25), 14863.

[10] Dadgostar, H. and Zarnegar, B. and Hoffmann, A. and Qin, X.F. and Truong, U. and Rao, G. and Baltimore, D. and Cheng, G. (2002). Cooperation of multiple signaling pathways in CD40-regulated gene expression in B lymphocytes. *Proceedings of the National Academy of Sciences*, **99**(3), 1497.

[11] de Hoon, M. and Imoto, S. and Miyano, S. (2002). A comparison of clustering techniques for gene expression data. in *Proc. of the 10th Intl Conf. on Intelligent Systems for Molecular Biology*.