Using viper, a package for Virtual Inference of Protein-activity by Enriched Regulon analysis

Mariano J. Alvarez[1,2], Federico M. Giorgi[1], and Andrea Califano[1]

[1]Department of Systems Biology, Columbia University, 1130 St. Nicholas Ave., New York
[2]DarwinHealth Inc, 3960 Broadway, New York

October 24, 2025

1 Overview of VIPER

Phenotypic changes effected by pathophysiological events are now routinely captured by gene expression profile (GEP) measurements, determining mRNA abundance on a genome-wide scale in a cellular population[8, 9]. In contrast, methods to measure protein abundance on a proteome-wide scale using arrays[11] or mass spectrometry[10] technologies are far less developed, covering only a fraction of proteins, requiring large amounts of tissue, and failing to directly capture protein activity. Furthermore, mRNA expression does not constitute a reliable predictor of protein activity, as it fails to capture a variety of post-transcriptional and post-translational events that are involved in its modulation. Even reliable measurements of protein abundance, for instance by low-throughput antibody based methods or by higher-throughput methods such as mass spectrometry, do not necessarily provide quantitative assessment of functional activity. For instance, enzymatic activity of signal transduction proteins, such as kinases, ubiquitin ligases, and acetyltransferases, is frequently modulated by post-translational modification events that do not affect total protein abundance. Similarly, transcription factors may require post-translationally mediated activation, nuclear translocation, and co-factor availability before they may regulate specific repertoires of their transcriptional targets. Finally, most target-specific drugs affect the activity of their protein substrates rather than their protein or mRNA transcript abundance.

The VIPER (Virtual Inference of Protein-activity by Enriched Regulon analysis) algorithm[12] allows computational inference of protein activity, on an individual sample basis, from gene expression profile data. It uses the expression of genes that are most directly regulated by a given protein, such as the targets of a transcription factor (TF), as an accurate reporter of its activity.

We have shown that analysis of TF targets inferred by the ARACNe algorithm[1, 13], using the Master Regulator Inference algorithm (MARINA)[4], is effective in identifying drivers of specific cellular phenotypes which could be experimentally validated[4, 6]. While VIPER exploits the same principle as MARINA, it implements a dedicated algorithm specially formulated to estimate regulon activity, which takes into account the regulator mode of action, the regulator-target gene interaction confidence and the pleiotropic nature of each target gene regulation. In addition, while especially straightforward for TFs, VIPER effectively extends to signal transduction proteins. For this, we extended the concept of regulon to include the transcriptional targets that are most directly affected by the protein's activity, based on maximization of information transfer over all alternative paths[12].

VIPER is provided in this package in two flavors: a multiple sample version (msVIPER) designed for gene expression signatures based in multiple samples or expression profiles, and the single sample version (VIPER), which estimates relative protein activity on a sample-by-sample basis, thus allowing transformation of a typical gene expression matrix (i.e. multiple mRNA profiled across multiple samples) into a protein activity matrix, representing the relative activity of each protein in each sample.

Table 1: Regulatory networks described in [12] and available from figshare.

Title	Figshare citation
Human B-cell transcriptional network	http://dx.doi.org/10.6084/m9.figshare.680885
Human B-cell transcriptional network	http://dx.doi.org/10.6084/m9.figshare.680888
Human glioma transcriptional network	http://dx.doi.org/10.6084/m9.figshare.680887
MCF7 human breast carcinoma cell line transcriptional network	http://dx.doi.org/10.6084/m9.figshare.680889
Human breast carcinoma signalome network	http://dx.doi.org/10.6084/m9.figshare.695962

The *viper* package implements VIPER and msVIPER algorithms in R. The *bcellViper* data package provides some example datasets and a small B-cell context-specific transcriptional regulatory network, representing 172,240 inferred regulatory interactions between 621 TFs and 6,249 target genes. Additional networks can be obtained from figshare (Table 1) and from the author's web site (`http://wiki.c2b2.columbia.edu/califanolab/index.php/Software`).

2 Citation

Alvarez MJ, Shen Y, Giorgi FM, Lachmann A, Ding BB, Ye BH & Califano, A. Functional characterization of somatic mutations in cancer using network-based inference of protein activity. Nature Genetics (In press) (2016).

3 Installation of *viper* package

Viper requires the R-system (`http://www.r-project.org`), the *mixtools* package (`http://www.bioconductor.org`), and the (*bcellViper*) package to run the examples. After installing R, all required components can be obtained with:

```
> if (!requireNamespace("BiocManager", quietly=TRUE))
+     install.packages("BiocManager")
> BiocManager::install("mixtools")
> BiocManager::install("bcellViper")
> BiocManager::install("viper")
```

4 Getting started

As first step, we have to load the viper environment with:

```
> library(viper)
```

5 Generating the *regulon* object

As described under 'Overview of VIPER' (section 1), msVIPER and VIPER require a gene expression signature and an appropriate cell context-specific regulatory network. This regulatory network is provided in the format of a class *regulon* object. Regulon objects can be generated from networks reverse engineered with the ARACNe algorithm [1]. This is performed by the function *aracne2regulon*, which takes two arguments as input: the ARACNe output *.adj* file, and the expression data-set used by ARACNe to reverse engineer the network. As an example, the package *bcellViper* provides a subset of the ARACNe output file containing the network for 20 TF regulators (*bcellaracne.adj* file). For convenience, the full network is also provided, as a *regulon* class object, together with the gene expression data used to reverse engineer it contained in an

EpressionSet object. The B-cell expression data contains 211 samples representing several normal and tumor human B-cell phenotypes profiled on Affymetrix H-GU95Av2 (Gene Expression Omnibus series GSE2350)[1]. The provided dataset was generated from custom probe-clusters obtained by the the cleaner algorithm[3] and MAS5[2] normalization.

The following lines are an example for the use of *aracne2regulon* function to generate the *regulon* object from the ARACNe output data and the expression data:

```
> data(bcellViper, package="bcellViper")
> adjfile <- system.file("aracne", "bcellaracne.adj", package = "bcellViper")
> regul <- aracne2regulon(adjfile, dset, verbose = FALSE)

> print(regul)

Object of class regulon with 20 regulators, 3758 targets and 6013 interactions
```

6 Master Regulator Analysis performed by msVIPER

To illustrate this section, we analyze part of the expression data from [1], consitent on 5 naïve human B-cell, 5 memory B-cell, 5 centroblast and 5 centrocyte B-cell samples profiled on Affymetrix H-GU95Av2 gene arrays. The complete dataset is available from Gene Expression Omnibus (GSE2350), and here for convenience, we have included the 'cleaner'[3] processed and MAS5[2] normalized samples in the *bcellViper* package.

6.1 Generating the gene expression signatures

Lets assume that we are interested in identifying transcriptional regulators associated with the Germinal Center (GC) reaction. GC are the peripheral lymphoid organs where antigen-driven somatic hypermutation of the genes encoding the immunoglobulin variable region occurs, and are the main source of memory B cells and plasma cells that produce high-affinity antibodies. Centroblast and centrocyte are the main B-cell phenotypes present in the GCs, they are derived from antigen-stimulated peripheral blood B-cells, and represent the most proliferative cellular physiologic phenotypes of the adult human body. Thus, we can obtain a gene expression signature for the GC formation by comparing GC (centroblasts and centrocytes) against naïve B-cells. The 'ExpressionSet' object available from the *bcellViper* data package contains 5 centroblast samples (CB), 5 centrocyte samples (CC) and 5 naïve peripheral blood B-cell samples (N).

The *viper* package includes the function *rowTtest* that efficiently performs Student's t-test for each row of a dataset. The *rowTtest* function conveniently takes an 'ExpressionSet' object as argument and produces a list object containing the Student's t-statistic (`statistic`) and the test's p-value (`p.value`), that by default is estimated by a 2-tail test.

```
> signature <- rowTtest(dset, "description", c("CB", "CC"), "N")
```

It can also take two matrixes as arguments, the first one containing the 'test' samples and the second the 'reference' samples.

While we could define the Gene Expression Signature (GES) by using the t-statistic, to be consistent with the z-score based null model for msVIPER (see section 6.2), we will estimate z-score values for the GES:

```
> signature <- (qnorm(signature$p.value/2, lower.tail = FALSE) *
+                 sign(signature$statistic))[, 1]
```

6.2 NULL model by sample permutations

A uniform distribution of the targets on the GES is not a good prior for msVIPER. Given the high degree of co-regulation in transcriptional networks, the assumption of statistical independence of gene expression

is unrealistic an can potentially lead to p-value underestimates. To account for the correlation structure between genes, we define a null model for msVIPER by using a set of signatures obtained after permuting the samples at random. The function *ttestNull* performs such process by shuffling the samples among the 'test' and 'reference' sets, according to the re-sampling mode and number of permutations indicated by the parameters *repos* and *per*, respectively.

```
> nullmodel <- ttestNull(dset, "description", c("CB", "CC"), "N", per = 1000,
+                        repos = TRUE, verbose = FALSE)
```

As output, the *ttestNull* function produces a numerical matrix of z-scores, with genes/probes in rows and permutation iterations in columns, than can be used as null model for the msVIPER analysis.

6.3 msVIPER

The last element required by msVIPER that we are still missing is an apropriate cell context-specific regulatory network. We have included a B-cell regulatory network in the *bcellViper* package, and additional networks described in [12] for human B-cell, glioma and breast carcinoma can be obtained from figshare (Table 1).

```
> regulon
```

```
Object of class regulon with 621 regulators, 6249 targets and 172240 interactions
```

The msVIPER analysis is performed by the *msVIPER* function. It requires a *GES*, *regulon object* and *null model* as arguments, and produces an object of class 'msVIPER', containing the GES, regulon and estimated enrichment, including the Normalized Enrichment Score (NES) and p-value, as output.

```
> mrs <- msviper(signature, regulon, nullmodel, verbose = FALSE)
```

The reults can be summarized by the generic function *summary*, which takes the msviper object and either the number of top regulators to report or a specific set of regulators to list. The default for this parameter is the top 10 master regulators (MRs).

```
> summary(mrs)
```

```
        Regulon Size   NES  p.value   FDR
TCF3       TCF3  298  3.30 0.000962 0.028
BCL6       BCL6  401  3.28 0.001030 0.028
KLF10     KLF10  254  3.25 0.001170 0.028
MYBL2     MYBL2  240  3.22 0.001300 0.028
TSC22D3 TSC22D3  333 -3.17 0.001520 0.028
HES1       HES1  360 -3.18 0.001470 0.028
ZNF32     ZNF32  291 -3.19 0.001420 0.028
ZMYND11 ZMYND11  452 -3.20 0.001380 0.028
ZNF101   ZNF101  301 -3.22 0.001300 0.028
KLF9       KLF9  337 -3.24 0.001190 0.028
```

A graphics representation of the results (msVIPER plot) can be obtained by the generic function *plot* (shown in Fig. 1). It takes the *msviper* object and either, the number of top differentially active regulators, or the names of the regulators to include in the plot as arguments. The default behavior is to plot the top 10 most differentially active MRs.

```
> plot(mrs, cex = .7)
```

6.3.1 Leading-edge analysis

msVIPER infers the relative activity of a regulatory gene based on the enrichment of its most closely-regulated targets on a given GES, but does not identify which are the target genes enriched in the GES. Subramanian et al. [14] proposed a method called leading-edge analysis to identify the genes driving the enrichment of a gene-set on a GES based on Gene Set Enrichment Analysis (GSEA). We implemented the leading-edge analysis in the *ledge* function of the *viper* package. The function only has a 'msviper' class object as argument and generates an updated 'msviper' object that now includes a 'ledge' slot.

```
> mrs <- ledge(mrs)
> summary(mrs)
```

```
        Regulon Size   NES  p.value   FDR
TCF3       TCF3  298  3.30 0.000962 0.028
BCL6       BCL6  401  3.28 0.001030 0.028
KLF10     KLF10  254  3.25 0.001170 0.028
MYBL2     MYBL2  240  3.22 0.001300 0.028
TSC22D3 TSC22D3  333 -3.17 0.001520 0.028
HES1       HES1  360 -3.18 0.001470 0.028
ZNF32     ZNF32  291 -3.19 0.001420 0.028
ZMYND11 ZMYND11  452 -3.20 0.001380 0.028
ZNF101   ZNF101  301 -3.22 0.001300 0.028
KLF9       KLF9  337 -3.24 0.001190 0.028
                                                Ledge
TCF3    SMARCA4, MCM7, TRAF3IP3, NDC80, + 110 genes
BCL6        KIF14, BUB1, DLGAP4, GINS1, + 217 genes
KLF10       TRIP13, NDC80, AHNAK, KIF2C, + 99 genes
MYBL2     SMARCA4, MCM7, TRIP13, GINS1, + 138 genes
TSC22D3      NOTCH2, RAD1, RBM19, MLEC, + 190 genes
HES1          CDK4, SHC1, STX7, MAN1A1, + 104 genes
ZNF32       GNA12, PLAG1, PSMB1, CARM1, + 145 genes
ZMYND11   ANKRD26, EXTL2, IGFBP4, CTSC, + 234 genes
ZNF101     SLC46A3, GCLM, TCEA2, HMOX2, + 128 genes
KLF9          IFIT1, LPAR1, NID1, STOM, + 158 genes
```

7 Beyond msVIPER

7.1 Bootstrap msVIPER

The effect of outlier samples on the gene expression signature can be reduced by the use of resampling techniques. msVIPER is capable of performing the analysis with bootstrap if a matrix of bootstraped signatures, instead of a vector, is given as *signature* argument. We implemened the function *bootstrapTtest* in the *viper* package to generate this kind of bootstraped GES matrixes from the 'test' and 'reference' datasets. The function produces 100 bootstrap interactions by default.

```
> signature <- bootstrapTtest(dset, "description", c("CB", "CC"), "N", verbose = FALSE)
> mrs <- msviper(signature, regulon, nullmodel, verbose = FALSE)
```

By default, *msviper* integrates the regulator activity results across all bootstraped iteration using the average, but this can be easily modified to use the median or mode values by the *bootstrapmsviper* function:

```
> mrs <- bootstrapmsviper(mrs, "mode")
```

Bootstraped msviper results can be displayed in the same way as non-bootstraped results (Fig. 2):

```
> plot(mrs, cex = .7)
```

7.2 Shadow analysis

A regulator may appear to be significantly activated based on its regulon's analysis, simply because several of its targets may also be regulated by a *bona fide* activated TF (shadow effect)[4, 5]. This constitutes a significant confounding issue, since transcriptional regulation is highly pleotropic, with individual targets being regulated by many TFs. msVIPER and VIPER (section 8) address this issue by penalizig the contribution of the pleotropically regulated targets to the enrichment score. However, a post-hoc shadow analysis, as described in [4] can still be applied to the msVIPER results with the function *shadow*. This function takes a class 'msviper' object, and performs a shadow analysis on a selected number of top MRs indicated by the argument *regulators*, which can be used to indicate either the enrichment p-value cutoff, the number of top MRs, or the names of the MRs to consider in the analysis.

```
> mrshadow <- shadow(mrs, regulators = 25, verbose = FALSE)
```

As output, the *shadow* function produces an updated 'msviper' object. The summary of it, generated by the *summary* function, lists now not only the top MRs, but also the shadow pairs, in the form: $MR_1 -> MR_2$, indicating that part of the inferred MR_2 ativity is due to co-regulation of MR_2 target genes by MR_1.

```
> summary(mrshadow)
```

```
$msviper.results
        Regulon Size   NES p.value    FDR
BCL6       BCL6  401  3.18 0.00146 0.0709
MYBL2     MYBL2  240  3.12 0.00184 0.0709
WHSC1     WHSC1  257  3.07 0.00213 0.0709
TOP2A     TOP2A  749  3.06 0.00222 0.0709
MYBL1     MYBL1  225  3.04 0.00239 0.0709
PTTG1     PTTG1  471  3.00 0.00273 0.0709
NR1D2     NR1D2  259 -3.00 0.00266 0.0709
TSC22D3 TSC22D3  313 -3.02 0.00255 0.0709
ZNF274   ZNF274  160 -3.04 0.00235 0.0709
ZMYND11 ZMYND11  452 -3.06 0.00220 0.0709

$Shadow.pairs
 [1] "BCL6 -> TCF3"      "BCL6 -> HES1"      "MYBL2 -> ZNF32"
 [4] "MYBL2 -> HES1"     "MYBL2 -> ZNF23"    "MYBL2 -> MEIS2"
 [7] "KLF10 -> ZNF101"   "KLF10 -> HES1"     "KLF9 -> HES1"
[10] "ZNF32 -> HES1"     "WHSC1 -> TSC22D3"  "WHSC1 -> ZNF101"
[13] "WHSC1 -> IRF5"     "WHSC1 -> KDM1A"    "WHSC1 -> E2F2"
[16] "TSC22D3 -> HES1"   "TSC22D3 -> ZNF23"  "TOP2A -> CREB3L2"
[19] "TOP2A -> E2F2"     "TOP2A -> HES1"     "TOP2A -> HMGB2"
[22] "TOP2A -> MEIS2"    "ZMYND11 -> IRF5"   "ZMYND11 -> E2F2"
[25] "ZMYND11 -> HES1"   "ZMYND11 -> MEIS2"  "CREB3L2 -> E2F2"
[28] "PRKDC -> HES1"     "IRF5 -> HES1"      "PTTG1 -> E2F2"
[31] "ZNF274 -> E2F2"    "ZNF274 -> ZNF23"   "ZNF274 -> MEIS2"
[34] "NR1D2 -> HES1"     "NR1D2 -> HMGB2"    "HES1 -> HMGB2"
[37] "WHSC1 -> KLF10"    "TSC22D3 -> KLF10"  "PRKDC -> KLF10"
[40] "ZNF274 -> KLF10"   "KDM1A -> KLF10"    "ZNF23 -> KLF10"
[43] "HMGB2 -> KLF10"    "MEIS2 -> KLF10"    "ZNF32 -> KLF9"
[46] "PRKDC -> KLF9"     "MYBL1 -> KLF9"     "IRF5 -> KLF9"
[49] "ZNF274 -> KLF9"    "WHSC1 -> ZNF32"    "TSC22D3 -> ZNF32"
[52] "KDM1A -> ZNF32"    "HMGB2 -> ZNF32"    "PRKDC -> TCF3"
[55] "MYBL1 -> TCF3"     "IRF5 -> TCF3"      "PTTG1 -> TCF3"
[58] "KDM1A -> TCF3"     "HMGB2 -> TCF3"     "TOP2A -> ZNF101"
```

```
[61] "PRKDC -> ZNF101"   "MYBL1 -> ZNF101"   "PTTG1 -> ZNF101"
[64] "ZNF274 -> ZNF101"  "NR1D2 -> ZNF101"   "PTTG1 -> CREB3L2"
[67] "ZNF274 -> CREB3L2" "HMGB2 -> CREB3L2"  "MYBL1 -> PRKDC"
[70] "HMGB2 -> MYBL1"    "ZNF274 -> PTTG1"   "HMGB2 -> KDM1A"
[73] "NR1D2 -> E2F2"
```

7.3 Synergy analysis

To predict synergistic interactions between regulators we first compute the enrichment of co-regulons, defined as the intersection between regulons. We expect that a combination of regulators will synergistically regulate a gene expression signature if their co-regulon show a significantly higher enrichment on the signature than the union of the corresponding regulons[6]. Co-regulon analysis is implemented in the *viper* package by the *msviperCombinatorial* function. It takes a 'msviper' object as argument and computes the enrichment of all co-regulons, generated from a selected number of MRs (indicated by the *regulators* parameter), on the GES. As an example, we compute the enrichment of the co-regulons for the top 25 regulators,

```
> mrs <- msviperCombinatorial(mrs, regulators = 25, verbose = FALSE)
```

The comparison between the enrichment of the co-regulon versus the union of the corresponding regulons (synergy analysis) is implemented by the function *msviperSynergy*, which requires only a 'msviper' object generated by *msviperCombinatorial* and the number of permutations used to compute the p-values, which default is 1,000:

```
> mrs <- msviperSynergy(mrs, verbose = FALSE)
```

The output of *msviperSynergy* is un updated object of class 'msviper' with *plot* (Fig. 3) and *summary* methods. The output of *summary* will include in this case the enrichment results for the co-regulons and the p-value for the predicted synergistic effect.

```
> summary(mrs)
```

	Regulon	Size	NES	p.value	FDR	Synergy
ZNF32--PRKDC	ZNF32--PRKDC	30	4.74	2.17e-06	0.00138	1.48e-06
ZMYND11--PRKDC	ZMYND11--PRKDC	38	4.46	8.22e-06	0.00138	6.55e-05
MYBL2--CREB3L2	MYBL2--CREB3L2	37	4.45	8.44e-06	0.00138	5.30e-06
MYBL2--TSC22D3	MYBL2--TSC22D3	35	4.45	8.52e-06	0.00138	7.21e-06
BCL6--ZNF32	BCL6--ZNF32	35	4.44	9.13e-06	0.00138	1.79e-05
ZNF32--CREB3L2	ZNF32--CREB3L2	27	4.34	1.45e-05	0.00146	4.83e-07
PRKDC--NR1D2	PRKDC--NR1D2	42	4.30	1.71e-05	0.00146	7.19e-07
KLF9--HES1	KLF9--HES1	41	4.28	1.87e-05	0.00146	1.43e-03
BCL6--TOP2A	BCL6--TOP2A	104	4.27	1.99e-05	0.00146	1.12e-06
TCF3--ZMYND11	TCF3--ZMYND11	31	4.26	2.03e-05	0.00146	3.95e-04

```
> plot(mrs, 25, cex = .7)
```

8 Virtual Inference of Protein-activity by Enriched Regulon analysis (VIPER)

VIPER is the extension of msVIPER to single sample-based analysis. It effectively transforms a gene expression matrix to a regulatory protein activity matrix. The simplest implementation of VIPER is based on single-sample gene expression signatures obtained by scaling the probes or genes – subtracting the mean and dividing by the standard devition of each row. A gene expression matrix or 'ExpressionSet' object and appropriate regulatory network are the minimum set of parameters required to perform a VIPER analysis with the *viper* function.

```
> vpres <- viper(dset, regulon, verbose = FALSE)
```

The *viper* function generates a matrix – or 'ExpressionSet' object in case an 'ExpressionSet' object is given as input – of regulator's activity, containing 621 regulators x 211 samples in our example.

```
> dim(vpres)
```

```
Features  Samples
     621      211
```

The differential activity of regulatory proteins between groups of samples, for example between germinal center B-cell and Naïve B-cells, can be obtained by any hypothesis testing statistical method, like for example the Student's t-test:

```
> tmp <- rowTtest(vpres, "description", c("CB", "CC"), "N")
> data.frame(Gene = rownames(tmp$p.value), t = round(tmp$statistic, 2),
+ "p-value" = signif(tmp$p.value, 3))[order(tmp$p.value)[1:10], ]
```

```
         Gene      t  p.value
TOP2A   TOP2A  20.77 2.35e-11
WHSC1   WHSC1  17.44 2.13e-10
MYBL2   MYBL2  16.85 3.25e-10
ZNF274 ZNF274 -16.59 3.96e-10
BCL6     BCL6  16.46 4.36e-10
ZNF23   ZNF23 -16.43 4.47e-10
ZNF32   ZNF32 -16.36 4.70e-10
MORC2   MORC2  16.27 5.03e-10
ZNF101 ZNF101 -16.07 5.87e-10
MYB       MYB  15.96 6.40e-10
```

8.1 Running VIPER with a null model

VIPER computes the normalized enrichment score (NES) analytically, based on the assumption that in the null situation, the target genes are uniformly distributed on the gene expression signature. Because the extensive co-regulation of gene expression taking place in the cell, this assumption never holds true, and this is the reason why a null model based on sample permutations is used in msVIPER to estimate NES. The same approach can also be used for VIPER, given that a set of samples is used as reference for the analysis. We can generate a set of GESs based on a set of reference samples, and the corresponding null model based on sample permutations, with the function *viperSignature*. It takes two matrixes as arguments, the first one containing the expression data for all the 'test' samples, and the second corresponding to the 'reference' samples. If an 'ExpressionSet' object is used as input, the 'reference' samples should be indicated and the function will consider all the remaining samples as 'test' ones. The number of permutations for the null model can be defined by the *per* argument, whose default value is 1,000.

```
> vpsig <- viperSignature(dset, "description", "N", verbose = FALSE)
> vpres <- viper(vpsig, regulon, verbose = FALSE)
```

Because VIPER expresses activity for all the regulatory proteins in the same scale – normalized enrichment score –, euclidean distance is an appropriate measure of similarity between samples and we can, for example, perform an unsupervised hierarchical cluster analysis of the samples in a similar way we would do it in the case of gene expression data (Fig. 4):

```
> pos <- pData(vpres)[["description"]] %in% c("M", "CB", "CC")
> d1 <- exprs(vpres)[, pos]
> colnames(d1) <- pData(vpres)[["description"]][pos]
> dd <- dist(t(d1), method = "euclidean")
> heatmap(as.matrix(dd), Rowv = as.dendrogram(hclust(dd, method = "average")), symm = T)
```

We have developed, and included in the *viper* package, a function to compute the similarity between the columns of a gene expression or VIPER-predicted activity matrix. It follows the same concept as the two-tail Gene Set Enrichment Analysis (GSEA)[7], but it is based on the aREA algorithm[12]. The *viperSimilarity* function takes an expression or activity matrix as input, and generates a matrix of similarity scores between sample pairs, in the form of a 'similarityDistance' class object.

```
> dd <- viperSimilarity(d1)
```

We can use the generic function *scale* to 'scale' the similary matrix in the rage [-1; 1], and the resulting matrix will be analogous to a correlation matrix. In this case, identical signatures will produce a similarity score equal to 1, while perfectly reversed signatures will produce similarity scores equal to -1. Orthogonal signatures will be characterized by similarity scores close to zero. As for other matrixes of similarity, the 'signatureDistance' class object can be transformed into a 'distance' class object with the method *as.dist*, which in turn can be used to perform, for example, cluster analysis of the samples (Fig. 5).

```
> heatmap(as.matrix(as.dist(dd)), Rowv = as.dendrogram(hclust(as.dist(dd),
+ method = "average")), symm = T)
```

References

[1] Basso, K. et al. Reverse engineering of regulatory networks in human B cells. Nat. Genet. 37, 382-90 (2005).

[2] Gautier, L., Cope, L., Bolstad, B. M., and Irizarry, R. A. 2004. affy—analysis of Affymetrix GeneChip data at the probe level. Bioinformatics 20, 3 (Feb. 2004), 307-315.

[3] Alvarez, M. J., Sumazin, P., Rajbhandari, P. & Califano, A. Correlating measurements across samples improves accuracy of large-scale expression profile experiments. Genome Biol. 10, R143 (2009).

[4] Lefebvre, C. et al. A human B-cell interactome identifies MYB and FOXM1 as master regulators of proliferation in germinal centers. Mol. Syst. Biol. 6, 377 (2010).

[5] Jiang, Z. & Gentleman, R. Extensions to gene set enrichment. Bioinformatics (Oxford, England) 23, 306-13 (2007).

[6] Carro, M. S. et al. The transcriptional network for mesenchymal transformation of brain tumours. Nature 463, 318-25 (2010).

[7] Julio, M. K. -d. et al. Regulation of extra-embryonic endoderm stem cell differentiation by Nodal and Cripto signaling. Development 138, 3885-3895 (2011).

[8] Klein, U. et al. Transcriptional analysis of the B cell germinal center reaction. Proc. Natl. Acad. Sci. USA. 100, 2639-44 (2003).

[9] Tothill, R. W. et al. Novel molecular subtypes of serous and endometrioid ovarian cancer linked to clinical outcome. Clin. Cancer Res. 14, 5198-208 (2008).

[10] Bozovic, A. & Kulasingam, V. Quantitative mass spectrometry-based assay development and validation: From small molecules to proteins. Clin. Biochem. 46, 444-455 (2012).

[11] Wolf-Yadlin, A., Sevecka, M. & MacBeath, G. Dissecting protein function and signaling using protein microarrays. Curr. Opin. Chem. Biol. 13, 398-405 (2009).

[12] Alvarez MJ, Shen Y, Giorgi FM, Lachmann A, Ding BB, Ye BH & Califano, A. Functional characterization of somatic mutations in cancer using network-based inference of protein activity. Nature Genetics (In press) (2016).

[13] Margolin, A. A. et al. ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. BMC Bioinformatics 7 Suppl 1, S7 (2006).

[14] Subramanian, A. et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proc. Natl. Acad. Sci. USA 102, 15545-50 (2005).

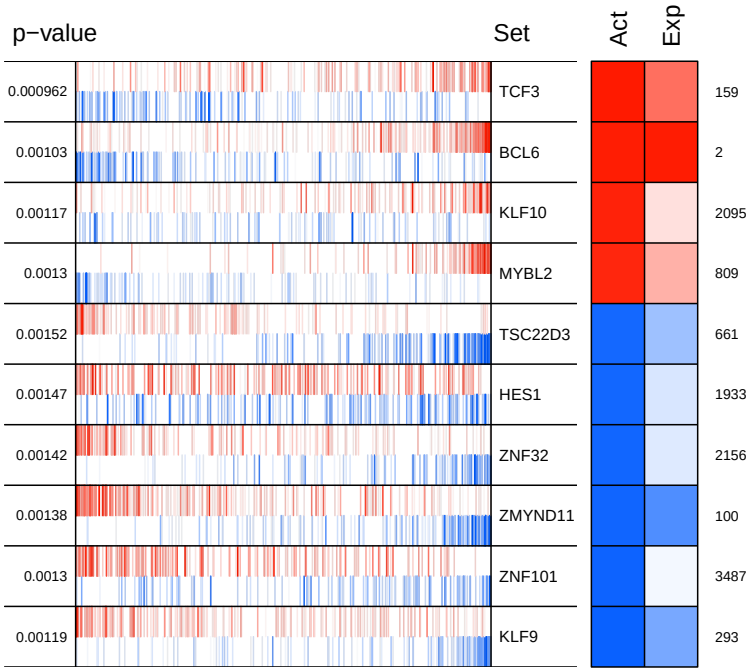

Figure 1: VIPER plot showing the projection of the negative (repressed, shown in blue color) and positive (activated, shown in red color) targets for each TF, as inferred by ARACNe and correlation analysis when reverse engineering the regulatory network (vertical lines resembling a bar-code), on the GES (*x-axis*), where the genes in the GES were rank-sorted from the one most down-regulated to the one most upregulated in the 'test' vs 'reference' conditions. The optional two-columns heatmap displayed on the right side of the figure shows the inferred differential activity (first column) and differential expression (second column), with the rank of the displayed genes in the GES (shown all the way to the right).

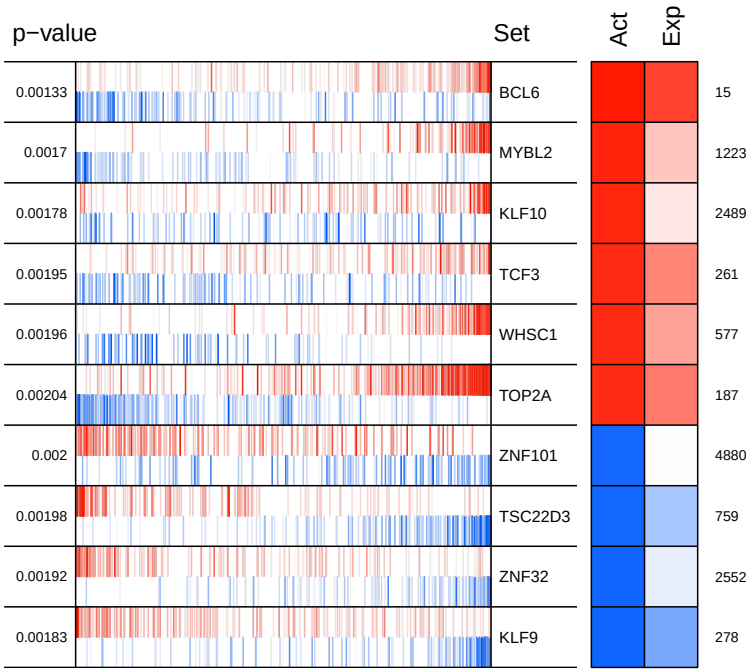

Figure 2: msVIPER plot showing the enrichment of transcription factor regulons on the germinal center reaction gene expression signature using 100 bootstrap iterations.

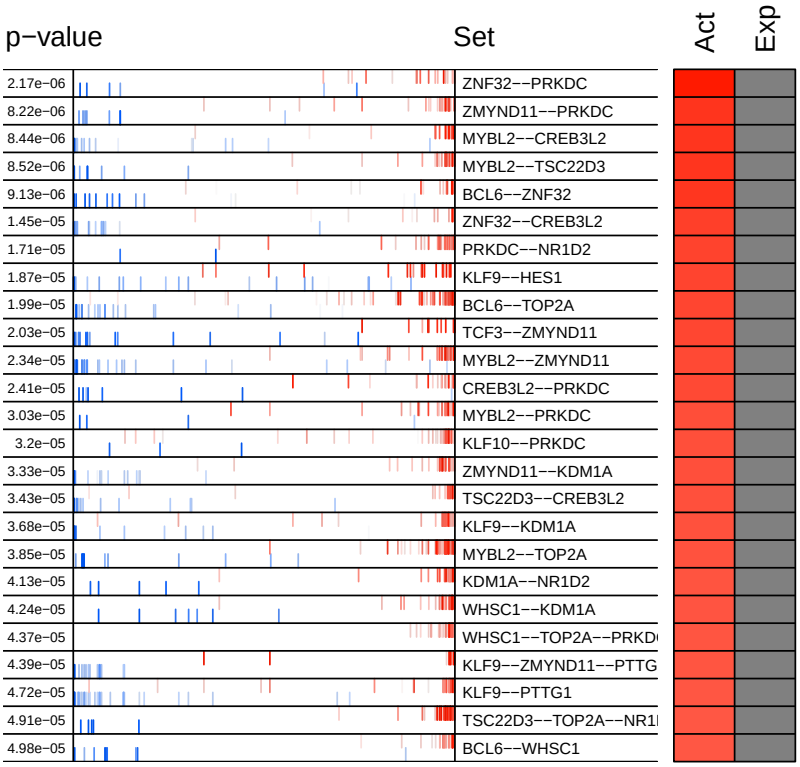

Figure 3: msVIPER plot showing the results for the enrichment of co-regulons on the germinal center reaction gene expression signature.

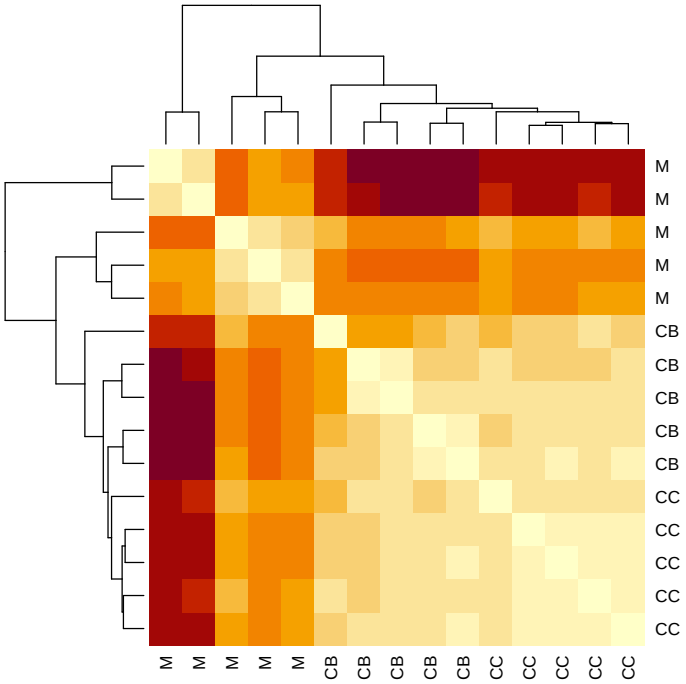

Figure 4: Heatmap showing the similarity between the samples (red indicated highly-similar samples) as measured by euclidean distance between the VIPER-inferred transcriptional regulator's activity profiles. The samples (M: memory B-cells, CB: centroblasts, CC: centrocytes) were arranged according to average-linkage hierarchical cluster analysis.

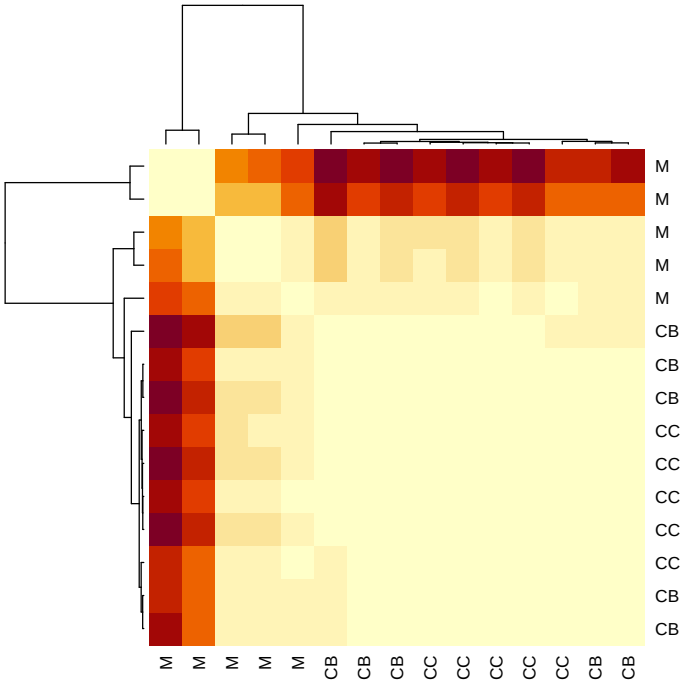

Figure 5: Heatmap showing the similarity between the samples (red indicated highly-similar samples) as measured by *viperSimilarity* between the VIPER-inferred regulatory protein activity profiles. The samples (M: memory B-cells, CB: centroblasts, CC: centrocytes) were arranged according to average-linkage hierarchical cluster analysis.