

# Using vulcan, a package that combines regulatory networks and ChIP-Seq data to identify chromatin-binding cofactors

Federico M. Giorgi<sup>1</sup>, Andrew N. Holding<sup>1</sup>, and Florian Markowetz<sup>1</sup>

<sup>1</sup>Cancer Research UK, Cambridge Institute, Robinson Way, Cambridge United Kingdom

October 29, 2019

## 1 Overview of VULCAN

Vulcan (VirtUaL ChIP-Seq Analysis through Networks) is a pipeline that combines ChIP-Seq data and regulatory networks to obtain transcription factors that are likely affected by a specific stimulus. In order to do so, our package combines strategies from different BioConductor packages: *DESeq* for data normalization, *ChIPpeakAnno* and *DiffBind* for annotation and definition of ChIP-Seq genomic peaks, *csaw* to define optimal peak width and *viper* for applying a regulatory network over a differential binding signature. Usage of gene regulatory networks to analyze biological systems has witnessed an exponential increase in the last decade, due to the ease of obtaining genome-wide expression data (Margolin et al., 2005; Giorgi et al., 2013; Castro et al., 2015). Recently, the VIPER approach to interrogate these network models has been proposed to infer transcription factor activity using the expression of a collection of their putative targets, i.e. their regulon (Alvarez et al., 2016). In the VIPER algorithm, gene-level differential expression signatures are obtained for either individual samples (relative to the mean of the dataset) or between groups of samples, and regulons are tested for enrichment. Ideally, TF regulons can be tested on promoter-level differential binding signatures generated from ChIP-Seq experiments, in order to ascertain the global change in promoter occupancy for gene sets. In our study, we propose an extension of the VIPER algorithm to specifically analyze TF occupancy in ChIP-Seq experiments. Our VULCAN algorithm uses ChIP-Seq data obtained for a given TF to provide candidate coregulators of the response to a given stimulus. The analysis is based on identifying differentially bound genes and testing their enrichment in the regulon of potential co-regulatory factors.

## 2 Citation

Giorgi FM, Holding AN & Markowetz F. Network Dynamics of early ER-alpha promoter binding in response to estradiol. *Genome Biology* (Submitted).

## 3 Installation of *vulcan* package

VULCAN requires the R-system (<http://www.r-project.org>), and the *vulcandata* 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("vulcandata")
> BiocManager::install("vulcan")
```

## 4 Getting started

As first step, we have to load the vulcan package with:

```
> library(vulcan)
```

Only to run the examples, a set of reduced BAM files can be accessed by loading the *vulcandata* package:

```
> library(vulcandata)
```

## 5 Running an example VULCAN pipeline

### 5.1 Generate a example annotation file

```
> # Generate an annotation file from the dummy ChIP-Seq dataset
> vfile<-tempfile()
> vulcandata::vulcansheet(vfile)
```

### 5.2 Import ChIP-Seq information

```
> # Import BAM and BED information into a list object
> # vobj<-vulcan.import(vfile)
> vobj<-vulcandata::vulcanexample()
> unlink(vfile)
```

### 5.3 Annotate peaks

```
> # Annotate peaks to gene names
> vobj<-vulcan.annotate(vobj,lborder=-10000,rborder=10000,method="sum")
```

### 5.4 Peak abundance normalization

```
> # Normalize data for VULCAN analysis
> vobj<-vulcan.normalize(vobj)
> # Detect which conditions are present in our imported object
> names(vobj$samples)
```

### 5.5 Load an ARACNe network

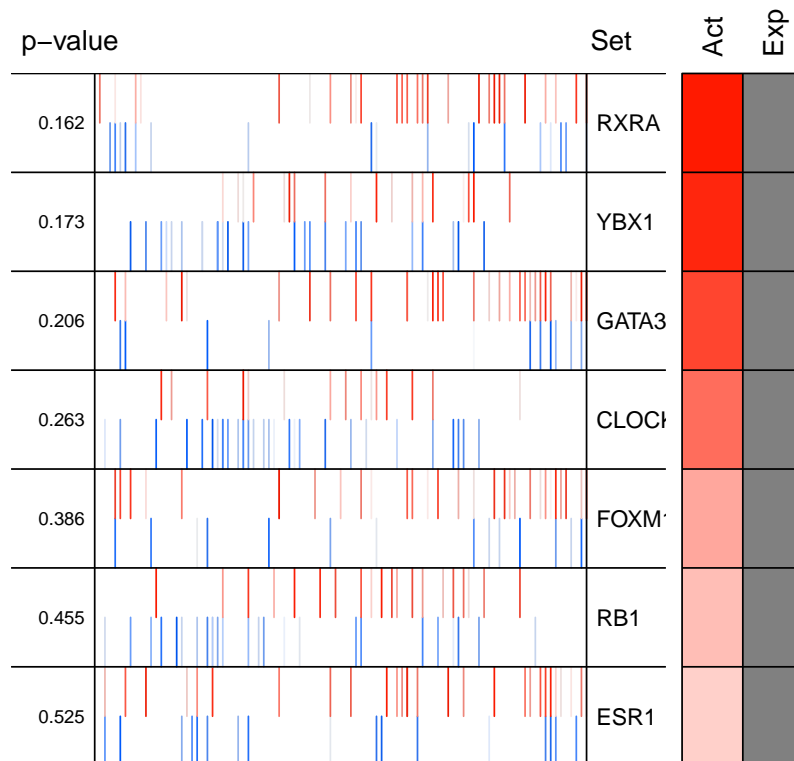
We will use a regulon object as specified in the *viper* package, named "network"

```
> load(system.file("extdata","network.rda",package="vulcandata",mustWork=TRUE))
```

### 5.6 Run VULCAN

Once we have prepared 1) the ChIP-Seq annotated and normalized object and 2) a gene regulatory network, we can run the VULCAN pipeline. We can reduce the minimum regulon size, since in the *vulcandata* example only one chromosome was used, and the networks would otherwise have too few hits. We can then visualize the output using the *msviper* plotting function

```
> vobj_analysis<-vulcan(vobj,network=network,contrast=c("t90","t0"),minsize=5)
> plot(vobj_analysis$msviper,mrs=7)
```



## 6 More convenience tools for ChIP-Seq functional data analysis

### 6.1 Gene Set Enrichment Analysis

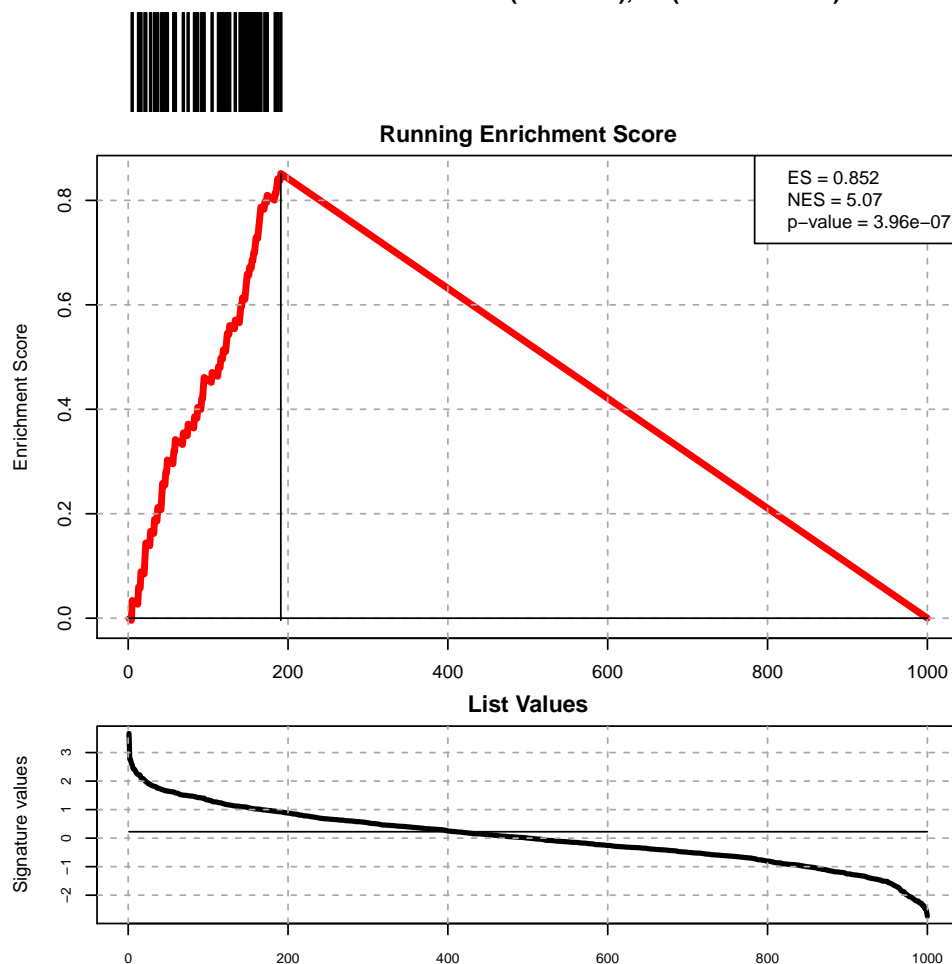
VULCAN contains a GSEA (Subramanian et al., 2005) implementation that can fit a Pareto tail distribution for p-value estimation. In the following example, we generate an artificial signature of 1000 genes with normally-distributed positive expression. The gene set that is evaluated is composed by 50 genes randomly picked from one of the tails

```
> reflist<-setNames(-sort(rnorm(1000)),paste0("gene",1:1000))
> set<-paste0("gene",sample(1:200,50))
> obj<-gsea(reflist,set,method="pareto")
> obj$p.value
```

The GSEA object can be plotted

```
> plot_gsea(obj)
```

Number of elements: 1000 (in full list), 50 (in element set)



## 6.2 Rank Enrichment Analysis

Rank Enrichment Analysis (REA Calculates) enrichment of groups of objects over a vector of values associated to a population of objects using a linear algebraic matrix multiplication.

```
> signatures<-setNames(-sort(rnorm(1000)),paste0("gene",1:1000))
> set1<-paste0("gene",sample(1:200,50))
> set2<-paste0("gene",sample(1:1000,50))
> groups<-list(set1=set1,set2=set2)
> obj<-rea(signatures=signatures,groups=groups)
> obj
```

## 6.3 Convenience functions

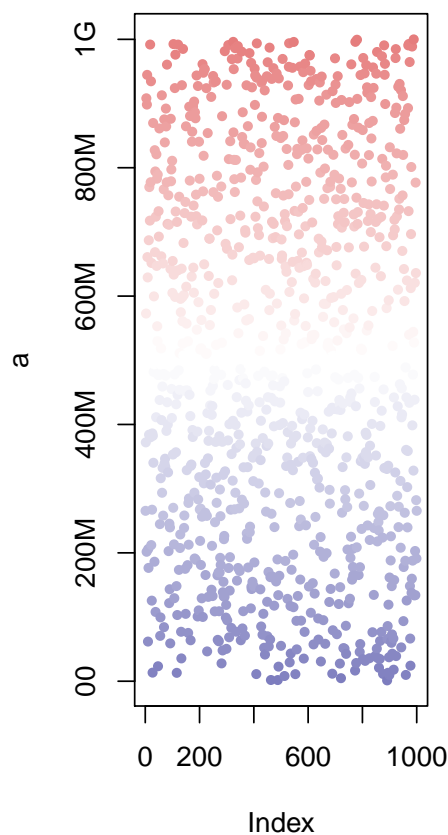
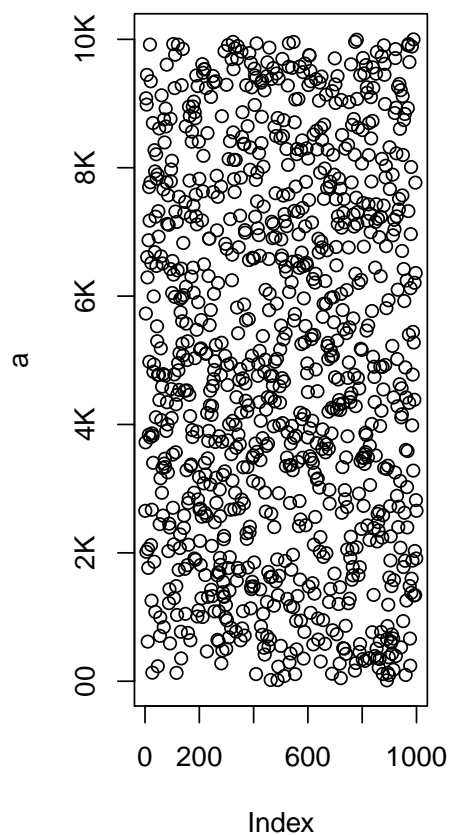
The function *kmgformat* will convert thousand numbers to K, millions to M, billions to G, trillions to T, quadrillions to P. The function *val2col* converts a distribution of numbers to a distribution of colors.

```
> par(mfrow=c(1,2))
> # Thousands
> set.seed(1)
```

```

> a<-runif(1000,0,1e4)
> plot(a,yaxt="n")
> kmg<-kmgformat(pretty(a))
> axis(2,at=pretty(a),labels=kmg)
> # Millions to Billions
> set.seed(1)
> a<-runif(1000,0,1e9)
> plot(a,yaxt="n",pch=20,col=val2col(a))
> kmg<-kmgformat(pretty(a))
> axis(2,at=pretty(a),labels=kmg)

```



## References

- [1] Alvarez, M.J., Shen, Y., Giorgi, F.M., Lachmann, A., Ding, B.B., Ye, B.H., & Califano, A. (2016). Functional characterization of somatic mutations in cancer using network-based inference of protein activity. *Nature Genetics*.
- [2] Margolin, A.A., Nemenman, I., Basso, K., Wiggins, C., Stolovitzky, G., Dalla Favera, R., & Califano, A. (2006). ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian

cellular context. *BMC bioinformatics*, 7(1), S7.

- [3] Castro, M.A., de Santiago, I., Campbell, T.M., Vaughn, C., Hickey, T.E., Ross, E., Tilley, W.D., Markowitz, F., Ponder, B.A. & Meyer, K.B., 2015. Regulators of genetic risk of breast cancer identified by integrative network analysis. *Nature genetics*.
- [4] Giorgi, F.M., Del Fabbro, C., & Licausi, F. (2013). Comparative study of RNA-seq-and microarray-derived coexpression networks in *Arabidopsis thaliana*. *Bioinformatics*, 29(6), 717-724.
- [5] Subramanian, A., Tamayo, P., Mootha, V.K., Mukherjee, S., Ebert, B.L., Gillette, M.A. & Mesirov, J.P. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences*, 102(43), 15545-15550.