TFEA.ChIP: a tool kit for transcription factor binding site enrichment analysis capitalizing on ChIP-seq datasets

Laura Puente-Santamaria, Luis del Peso

Introduction

The identification of the transcription factor (TF) responsible for the coregulation of an specific set of genes is a common problem in transcriptomics. In the most simple scenario, the comparison of the transcriptome of cells or organisms in two conditions leads to the identification of a set of differentially expressed (DE) genes and the underlying assumption is that one or a few TFs regulate the expression of those genes. Traditionally, the identification of the relevant TFs has relied on the use of position weight matrices (PWMs) to predict transcription factor binding sites (TFBSs) proximal to the DE genes (Wasserman and Sandelin, 2004). The comparison of predicted TFBS in DE versus control genes reveals factors that are significantly enriched in the DE gene set. The prediction of TFBS using these approaches have been useful to narrow down potential binding sites, but can suffer from high rates of false positives. In addition, this approach is limited by design to sequence-specific transcription factors (TF) and thus unable to identify cofactors that bind indirectly to target genes. To overcome these limitations we developed the R package TFEA.ChIP, which exploits the vast amount of publicly available ChIP-Seq datasets to determine TFBS proximal to a given set of genes and computes enrichment analysis based on this experimentally-derived rich information. Specifically TFEA.ChIP, uses information derived from the hundreds of ChIP-Seq experiments from the ENCODE Consortium[1] expanded to include additional datasets contributed to GEO database[2][3] by individual laboratories representing the binding sites of factors not assayed by ENCODE. The package includes a set of tools to customize the ChIP data, perform enrichment analysis and visualize the results. The package implements two enrichment analysis methods:

- Analysis of the association of TFBS and differential expression from 2x2 tables recording the presence of binding sites for a given TF in DE and control genes. The statistical significance of the association for each factor determined by a Fisher's exact test.
- GSEA analysis, based on the core function of the GSEA algorithm for R[4][5], *GSEA.EnrichmentScore*.

Building our TFBS database

The first source of ChIP-Seq datasets is Encode Uniform TFBS database, which guarantees a standard procedure to gather, filter, and share information from ChIP-Seq experiments. However, taking into consideration that the current estimations of the amount of transcription factors in the human genome range from 1,391[6] -manually curated candidates- to 2886 -predicted through computational methods by DBD[7][8]-, the 157 transcription factors covered by Encode's database were not considered enough to build a comprehensive TF enrichment analysis tool.

In order to expand the scope of our TFBS database, we also included datasets from ChIP-Seq experiments stored in GEO. In total, 1122 ChIP-Seq datasets, 689 from Encode and 433 from GEO DataSets, make up the source of information to generate this database, covering 333 different transcription factors in a variety of cell types and experimental conditions.

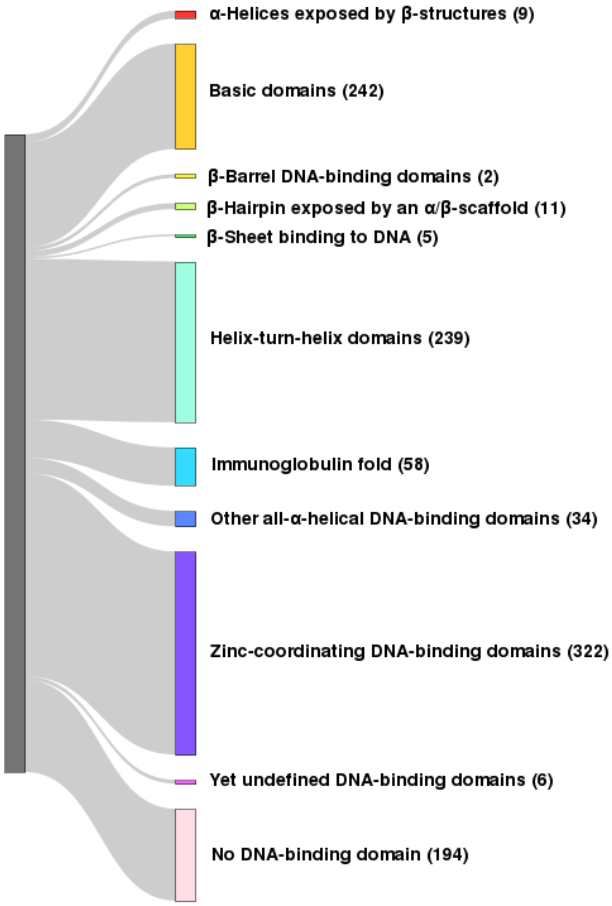


Illustration 1: Structural diversity according to DNA binding domains of the 333 transcription factors included in the TFBS database.

The process to establish a link between a peak in a ChIP-Seq experiment and a specific gene goes as follows:

- Generating a Dnase Hypersensitive Sites database, linking each Dnase HS to the nearest gene of those included in UCSC Known Gene database[9]. During this process, DHSs that were farther than 1Kb from any gene were discarded, so as to avoid highly uncertain connections that would undermine the robustness of any analysis. In the case of a DHSs close enough to more than one gene, both were assigned to the site. For this purpose we used Encode's Master DNaseI HS database[10][11].
- Selecting from each ChIP-Seq dataset those peaks that overlap a DHSs. Each of these peaks will be assigned the same gene as the DHS they overlap.
- Storing the list of genes assigned to a peak in each of the ChIP-Seq experiments. With this lists we generated a binary matrix which rows correspond to all the human genes in the Known Gene database, and its columns, to every ChIP-Seq experiment in the database; the values are 1 – if the ChIP-Seq has a peak assigned to that gene – or 0 – if it has not –.

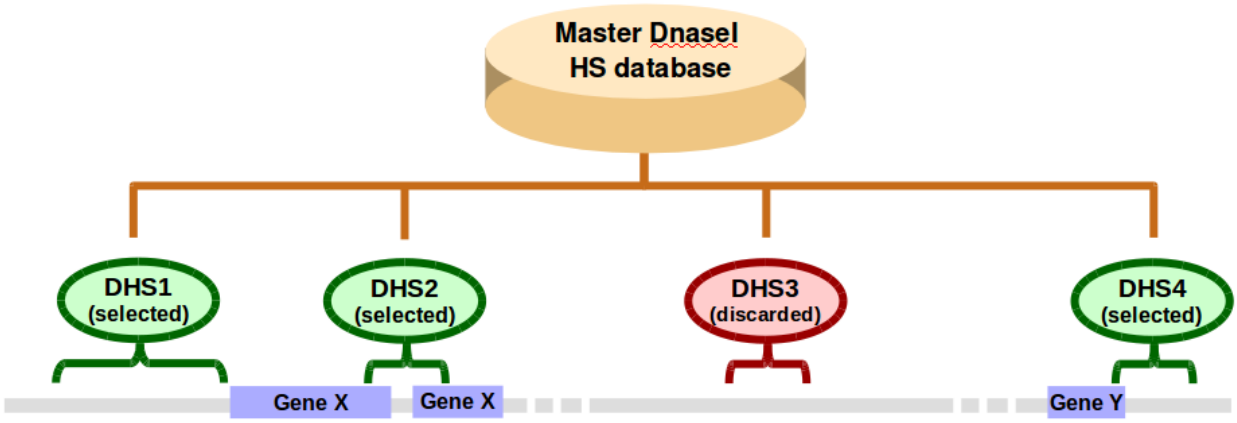


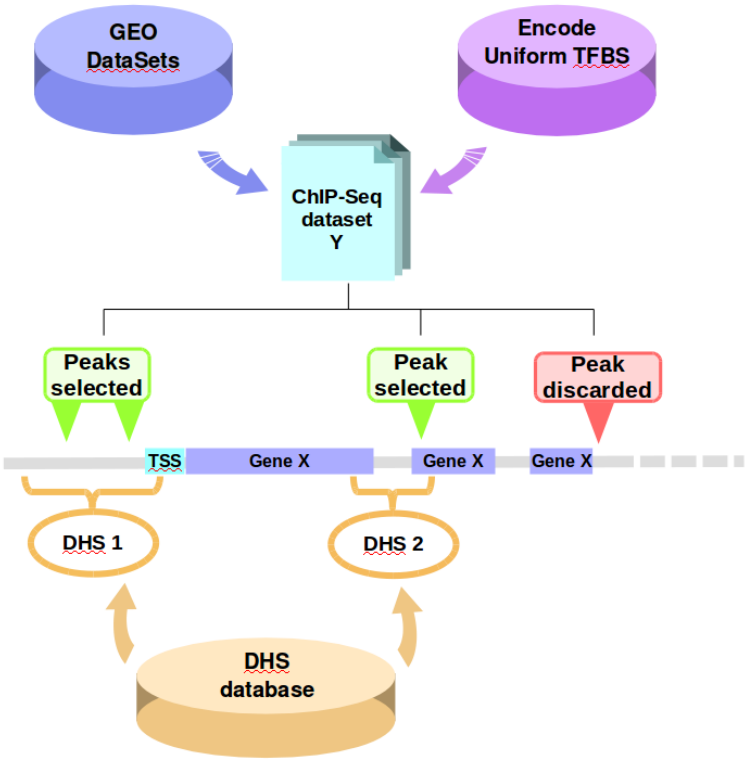
Illustration 2: Building a Dnase Hypersensitive Sites database. From Encode's Master DNaseI HS database, that gathers all DHSs found in several cell lines, we selected those DHSs that are 1Kb or closer to a gene according to the gene's location on UCSC's Known Gene database. DHS further than 1Kb from any gene are discarded to avoid highly uncertain links.

In this illustration, both DHS1 and DHS2 would be assign to gene "X", and DHS4 to gene "Y", while DHS3 would be discarded.

Illustration 3: Building the TFBS database.

For every peak in a ChIP-Seq dataset is tested whether it overlaps any of the DHSs located close to a gene. If the result is positive, the gene corresponding to said DHS –gene "X" in this illustration– will be assigned to the ChIP-Seq experiment, if its negative, that peak will be discarded.

The ChIP-Seq dataset "Y" has three peaks that overlap DHS1 and DHS2, so the Entrez ID of gene "X" would be associated to the ChIP-Seq "Y".



Analyzing TF enrichment with TFEA.ChIP

Input data

TFEA.ChIP is designed to take the output of a differential expression analysis and identify transcription factors enriched in the list of differentially expressed genes. The core premise of our method is that key effectors of a regulatory response will have more target genes among the differentially expressed than among the unresponsive genes.

In the case of the analysis of association, the only required input is a set of DE genes and, optionally, a set of control genes whose expression is not altered by the experimental conditions under the study.

For the GSEA analysis a ranked list of genes is required. This is supplied as a matrix or data frame containing a column with gene names and a numerical column with the ranking metric, which typically are log2(Fold change) for the gene expression changes in the two conditions under evaluation.

We will derive the input required for both analysis from a DESeq2[12] Results objects or a table containing the following field columns:

- Gene name (*Genes*). the program uses Entrez IDs, but translating from Gene Symbols or Ensembl IDs is available.
- Log2 Fold Change (*log2FoldChange*) indicating the difference between two experimental conditions (i.e.: normoxia vs. hypoxia, or control vs. TNF addition).
- P-value or adjusted p-value (*pvalue or pval.adj*) for the difference in a genes expression between the two conditions.

Genes	UNTR_avg	TNF_avg	log2FoldChange	pvalue
A1BG	0.227849	0.1512	-0.591627	0.468529
A1BG-AS1	0.137478	0.158948	0.209352	0.804791
A1CF	0	0.00235745	1.79769E+308	0.340684
A2LD1	0.0156528	0.0340409	1.12085	0.526962

Table 1.: Example of an input table for enrichment analysis. The required fields must be named as is shown: Gene, log2FC, and pVal.

In order to make the preparation of the input data easier, the package includes the function *preprocessInputData()*, that takes as input a DESeq2[12] Results object or a data frame. This function will translate gene IDs to Entrez Gene ID format, sort the elements of the input according to decreasing log2(Fold Change) and adjust p-values if needed.

The user is able to select a specific set of TFs or ChIP-Seq experiments to perform the analysis or use the whole database using the function *get_chip_index*. This *chip_index* is a data frame containing the accession IDs and TF tested of each ChIP-Seq experiment and its task is to set what ChIPs should be use for an analysis.

The user can, for instance, select experiments from Encode project only:

```
chip_index<-get_chip_index(encodeFilter = TRUE)
```

Also, select experiments for certain transcription factors of interest:

```
chip_index<-get_chip_index(TFfilter = c("EPAS1","HIF1A","ARNT"))
```

Alternatively, this process could be done manually filtering the metadata database ("MetaData" in the package) if the user wants a more specific selection, such as ChIP-Seq experiments done in a particular cell type. The resulting *chip_index* can be used as an input variable in the functions *contingency_matrix*, *getCMstats*, and, *GSEA_run*.

Analysis of the association of TFBS and differential expression

The simplest approach to transcription factor enrichment consist on comparing how many targets of a given transcription factor are in two lists of genes. This is the course of action taken in this method, focused on finding differences in transcription factor enrichment between two distinct gene lists, one of differentially expressed genes (be it up-regulated, down-regulated or both) and a control group.

The first step in this method is dividing the genes in the input table into groups depending on their differential expression. The user can divide the genes as they see fit best for their dataset, but, as a general recommendation, we suggest to distribute them in three groups:

- Upregulated genes: p-value ≤ 0.05 and log2FC > 0
- Downregulated genes: p-value ≤ 0.05 and log2FC < 0
- Control genes: p-value > 0.85 and, if the group is too large, log2FC between 0.5 and -0.5.

If a control group is not provided, the function will generate one, consisting on all of the genes in UCSC's Known Gene database that aren't present in a given test group.

TFEA.ChIP works with Entrez gene IDs. If the user dataset has a different identification system, such as Gene Symbols or Ensembl gene IDs, the function *GeneID2entrez* will translate them.

Then, running the function *contingency_matrix* will generate contingency matrix for every ChIP-Seq experiment in the database, counting how many genes it interacts with in the test and control groups.

GSM1443809		
	+	-
Test	1195	507
Control	4378	3336

Table 1: Example of a contingency matrix done by TFEA.ChIP.

The next step is performing Fisher's exact test to check if the difference distribution is significant or not. After this, the output table is generated with g*etCMstats* . The table contains the following information:

- Experiment accession ID.
- TF tested.
- Odds ratio.
- -log(p-value).

Accession	-log(Pval)	OR	TF
wgEncodeEH002086	1.420	1.226	EP300
wgEncodeEH002085	32.112	0.645	KDM5B
wgEncodeEH002288	12.063	1.346	ATF3
wgEncodeEH002323	5.072	0.849	CREB1

Table 2: Example of the output of getCMstats.

Gene Set Enrichment Analysis

This method is based on the same principle than GSEA[4][5]: having the list of genes of an RNA-seq experiment ordered by log2(Fold Change), for every ChIP-Seq in the database, it checks if every gene interacts with the TF tested in every ChIP-Seq experiment: if it does, it adds a quantity, if it doesn't, it subtracts the same quantity. This method allows to estimate were the targets of a transcription factor are located in our log2(Fold Change) scale (up/down-regulated) and if those genes are condensed together or dispersed through the gene list.

The minimal inputs needed to run *GSEA_run* are a sorted gene array and its corresponding log2(Fold Change) vector. As it is usual to find genes in RNA-Seq experiments that are not included in the Known Gene database, such as RNA genes, the input has to be filtered before running *GSEA_run* to avoid artifacts.

In TFEA.ChIP, *GSEA_run* generates a vector called Running Enrichment Score (RES) for every ChIP-Seq, that starts and ends in 0 and has a range between 1 and -1, thus being a normalized parameter.

The output of this method is a list containing:

- Enrichment table, that stores:
  - Accession of the ChIP-Seq experiment.
  - Transcription factor
  - Enrichment score (ES), the maximum absolute value of the Running Enrichment Score.
  - P-value of the enrichment score (adjusted using FDR).
  - Argument: position on the gene list where the ES is reached

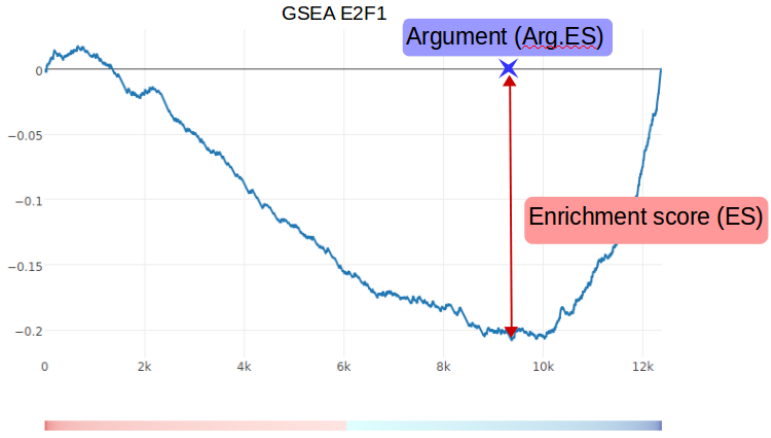


Illustration 4: This is an example of the output generated using one ChIP-Seq experiment on E2F1 and an RNA-Seq experiment on HUVEC cells in conditions of normoxia and hypoxia. As seen in the illustration, E2F1 has an ES of -0.2 and a high argument, meaning most of its targets are among the down-regulated genes.

- Running Enrichment Scores (RES): a numerical vector for each ChIP-Seq experiment storing the enrichment score at every element of the gene list.
- Indicator: a binary vector that stores whether every element of the gene list matched those genes assigned to the ChIP-Seq experiment or not.

Storing the RES and indicator vectors of ChIP-Seqs is optional. It's also possible -and recommended due to output size- to store only the RES and indicators of selected ChIP-Seq experiments.

Choosing a method

Both methods presented here offer a slightly different perspective on transcription factor enrichment. This difference presents the opportunity to analyze TF enrichment from several points of view, from the global picture of a process to the individual role of a specific transcription factor. Because of this, each of the method has distinct strengths depending on the data set used as an input.

- Timecourse experiments will benefit from an analysis of association, as it allows to integrate the information from all time steps at once, while a GSEA-like approach only admits data from a single time-step.
- Since the association method only uses differentially expressed genes –those which variation between experimental conditions is significantly different– it's the recommended one to use first in case the process being studied is not well known yet. In this way, the conclusions inferred from the analysis will be more robust.
- Experiments done with only one sample per condition shouldn't be analyze through the association method, as it needs genes that have significantly distinct averages between the two experimental conditions in the control and test groups. Dividing the genes by their fold change without taking into account the p-value of the measure can lead to weak results, as they might be based on a large proportion of genes incorrectly classified as up/down-regulated.
- The GSEA-like method, though slower, offers a more potent analysis, allowing the user not only to study the global behavior of TFs in a given condition, but to explore the interaction between specific transcription factors and every gene in their dataset.

Dealing with ±Infinite log2(FoldChange) values

One of the issues that may arise using expression data –particularly RNASeq– is interpreting the change in expression of a gene when it's not detected in all the conditions tested.

It's expected that some of these are, in fact off/on genes, but it may also an issue related to sequencing depth or other technical parameters. In part of those cases, there's not enough

certainty to conclude that the gene is only expressing in one condition, and, in addition to that, raw Fold Change will have a value of ±Inf, no matter the increase/decrease in expression.

In regard to transcription factor enrichment analysis, the specific log2(Fold Change) value is irrelevant for association analysis, but it can have a great impact in GSEA-like analysis, since GSEA's results are conditioned by the order of the genes provided as input.

Our suggestion is using tools for differential expression analysis, such as DESeq2[12], that have a mechanism to avoid infinite log2(Fold Change) values –usually, by adding pseudocounts to zero values–. If this is not possible, we strongly recommend to at least filter out those genes that have an infinite log2(Fold Change) but not a significant p-value.

Visualizing the results

Representing biological information in the present context of ever-growing complexity is a challenge for every group either developing or making use of bioinformatic tools. One of the goals for TFEA.ChIP is to offer a simple way to show the information generated after an enrichment analysis, offering not only statistical data –such as odds ratios or enrichment scores– but also the context for every ChIP-Seq experiment, since cell line or treatment can have a profound effect on the genes a TF is bound to on a particular situation.

For this reason, TFEA.ChIP incorporates several functions, based on the R package *plotly*[13], that generate interactive plots from the outputs of g*etCMstats* and *GSEA_run*. This allows to create interactive graphics in HTML format in which the user can zoom in/out and get extra information about a particular marker in the plot, thus easing data visualization.

plot_CM

The function *plot_CM* plots the output of *getCMstats*. To facilitate the identification of a subset of ChIP-Seq experiments of interest, *plot_CM* gives the option to highlight a set of markers by their transcription factor.

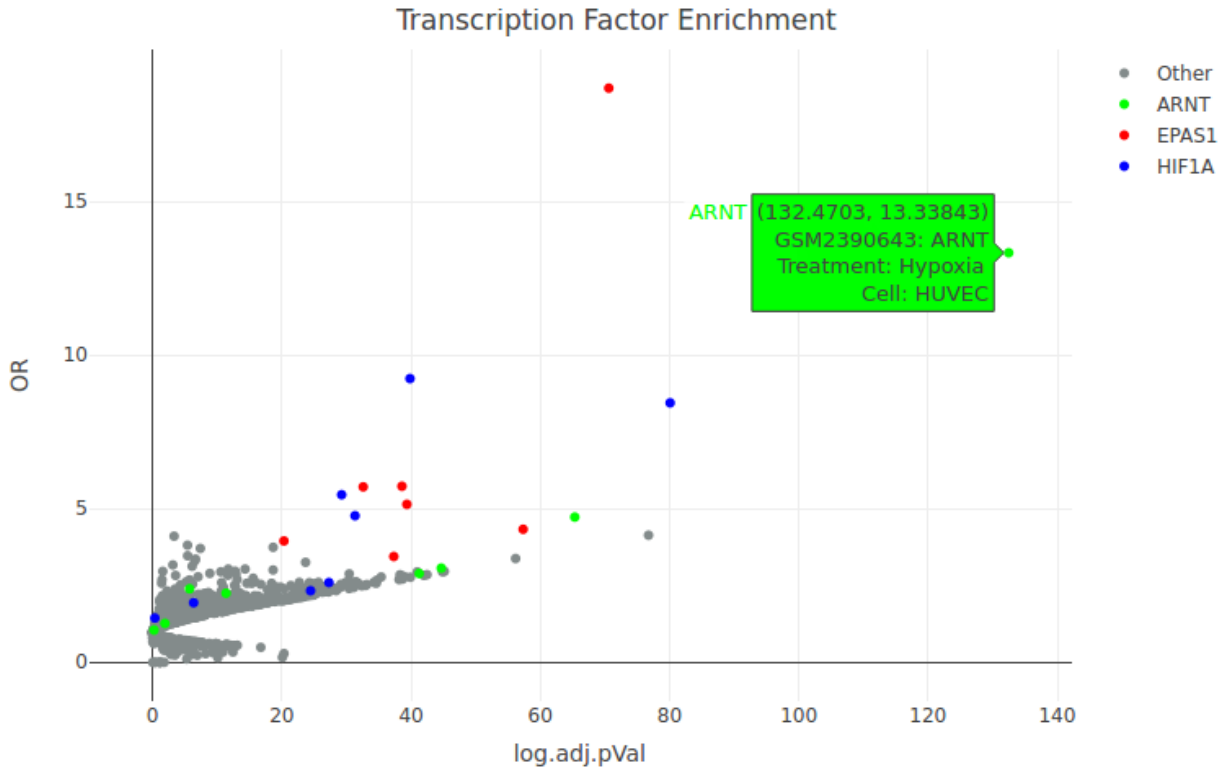


Illustration 5: Plot generated from an hypoxia vs. normoxia RNA-Seq experiment analyzing association of TFBS and differential expression. When the user positions the mouse over a marker, its metadata will appear on the screen.

As expected in this biological context, the subset of upregulated genes is highly enriched in HIF factors (HIF1A, EPAS1 and ARNT).

plot_ES

As can be seen in Illustration 6, *plot_ES* generates plots from *GSEA_run* outputs in a similar manner to *plot_CM*. In Enrichment Score plots, ES is represented against the argument of each ChIP-Seq (the element of the input gene list where the maximum enrichment score is reached). On the other hand the marker represents significance of the measurement, with an **x** for those ChIP-Seqs with a p-value under 0.05, and a ● for the rest.

Under the main plot, the color bar represents the log(Fold Change) distribution of the input gene list, the up-regulated genes in red, the down-regulated in blue.

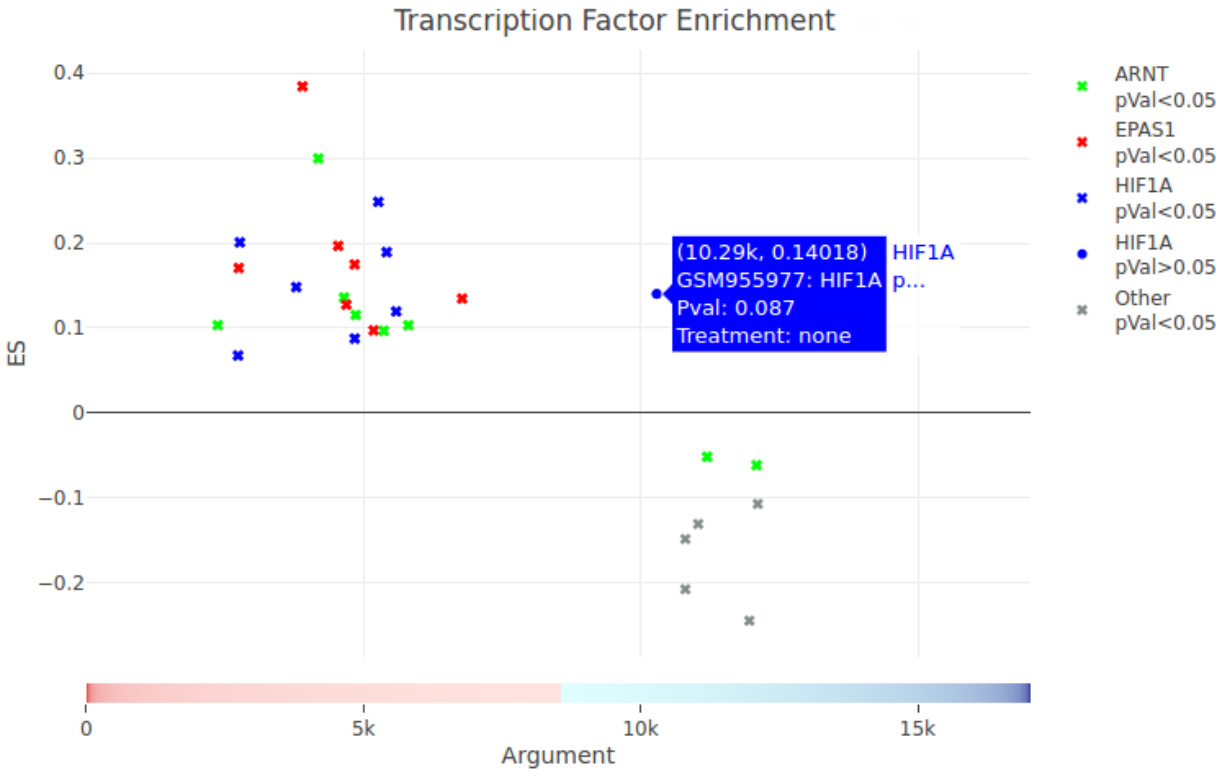


Illustration 6: Plot generated from an hypoxia vs normoxia RNA-Seq experiment using GSEA-like analysis.

In this plot, the Enrichment Score for every ChIP-Seq experiment is represented in the Y axis, while the X axis indicates the position on the gene list on which the Enrichment Score is reached.

In this type of plot p-value is represented using different markers:

- *x for p-val ≤ 0.05*
- *● for p-val > 0.05*

This plot was generated using an RNA-Seq experiment with HUVEC cells in two conditions, normoxia and hypoxia, after 8 hours. As expected, most of the ChIP-Seq experiments on HIF factors –HIF1A, EPAS1 and ARNT– have a positive ES, and their arguments are on the lower half of the scale, meaning that most of HIF's targeted genes are amongst the upregulated ones.

plot_RES

The package also includes the function *plot_RES* to generate plots from running enrichment scores. Using the variables *Accession* and *TF* the user can select which RES of those stored on their *GSEA_run* output to plot.

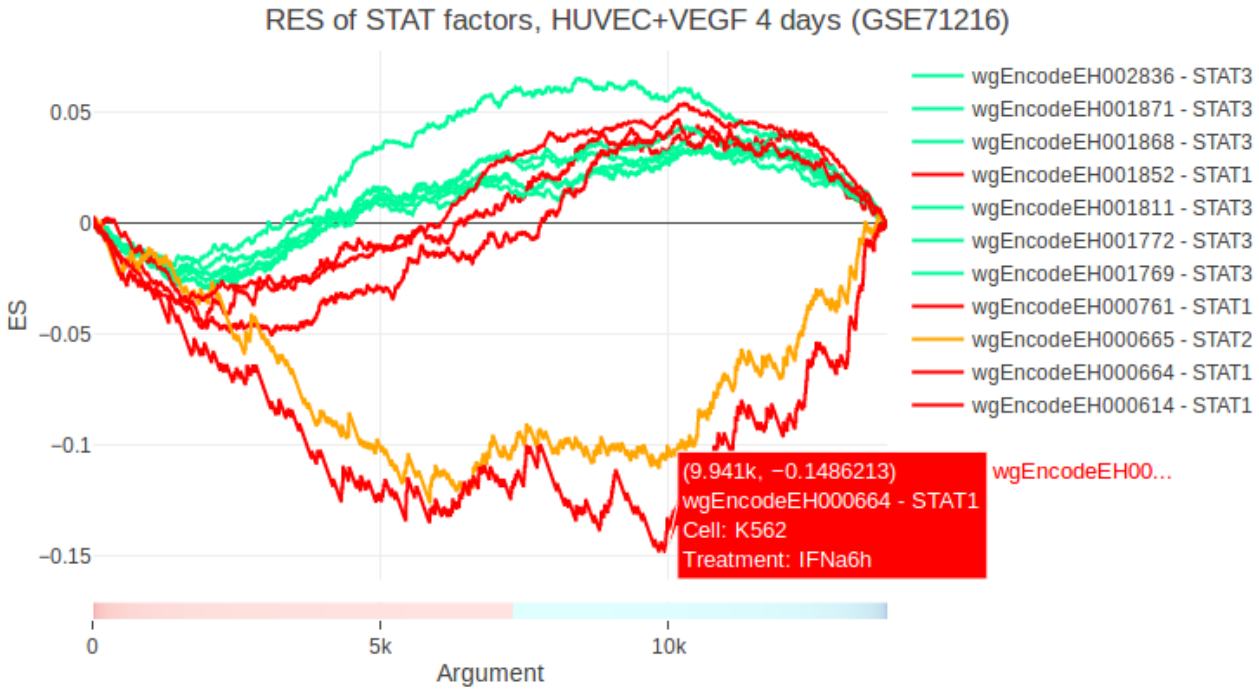


Illustration 7: plot of the Running Enrichment Score in an RNA-seq experiment on HUVEC cells supplemented with VEGF for 4 days for STAT1, STAT2 and STAT3 ChIP-Seqs.

The STAT1 ChIP-Seq wgEncodeEH000664 and the STAT2 ChIP-Seq were done with a IFNα treatment while the rest of STAT1 ChIPs were done using IFNγ. Since IFNα is a known angiogenesis inhibitor, the targets of this two ChIPs are significantly different than those of the ChIPs done with IFNγ, and in this case, amongst the downregulated genes.

Building your own TFBS database

If the user wants to build their TFBS database with different parameters, or add their own ChIP-seq datasets to use them along with the rest of the existing database, the functions *txt2GR, GR2tfbs_db*, and *makeTFBSmatrix* automate most of the process.

Starting data

- A Metadata table (storing at least, Accession ID, name of the file, and TF tested in the ChIP-Seq experiment). The metadata table included with this package has the following fields: "Name", "Accession", "Cell", "Cell Type", "Treatment", "Antibody", and "TF".
    - Name: name of the dataset file, e.g. “SMAD2_peaks.bed”.
    - Accession: accession ID of the experiment.
    - Cell: cell line or tissue in which the ChIP-Seq was performed.
    - Cell Type: more information about the cell, if needed.
    - Treatment: any non-standard condition in the experiment, e.g. addition of calcitriol, hypoxia, heat shock, etc.
    - Antibody.
    - TF: transcription factor tested in the ChIP-Seq experiment.
- A folder containing ChIP-Seq peak data, either in ".narrowpeak" format or the MACS output files "_peaks.bed" -a format that stores "chr", "start", "end", "name", and "Q-value" of every peak-.
- A Dnase Hypersensitive Site database. We used UCSC Master DNaseI HS, which includes all DHS from several cell lines, and discard from it the DHS that were too far from a gene to avoid uncertain links.

Step-by-step process

I. Filter the peaks of your datasets and convert them to GenomicRanges objects with *txt2GR*.

II. Load the Dnase Hypersensitive Sites database and generate your TFBS database with *GR2tfbs_db*.

III. Generate a binary matrix to use with the rest of TFEA.ChIP using *makeTFBSmatrix*. This matrix and the metadata table are the files you will need to run TFEA.ChIP with your own data.

IV. At the beginning of a session, use the function *set_user_data* to use your TFBS binary matrix and metadata table with the rest of the package.

Adding new datasets to existing TFBS matrix and metadata table

After generating a TFBS binary matrix or vector with your ChIP-Seq datasets, attach it to the existing TFBS matrix using cbind(). Add your ChIP-Seq’s metadata to the existing metadata table with rbind().

In order to keep using these new objects, save them and at the start of another session run the function *set_user_data* to temporarily substitute the package datasets.

Dependencies

The following R packages are used in TFEA.ChIP:

- Packages part of Bioconductor[14]:
  - GenomicRanges, IRanges, and GenomicFeatures[15]
  - biomaRt[16][17]
  - TxDb.Hsapiens.UCSC.hg19.knownGene[18]
  - org.Hs.eg.db[19]
- plotly[13]
- dplyr[20]
- knitr[21]
- rmarkdown[22]
- S4Vectors[23]
- scales[24]

Bibliography

1: ENCODE Project Consortium. Nature, 489:57-74

2: Edgar, R et al., Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. Nucleic Acids Res., 30:207-210

3: Barrett, T et al., NCBI GEO: archive for functional genomics data sets--update. Nucleic Acids Res., 41(Database issue):D991-995

4: Subramanian, Tamayo, et al., Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. PNAS, 102:15545-15550

5: Mootha, Lindgren, et al., PGC-1alpha-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. Nat Genet, 34:267-273

6: Vaquerizas et al., A census of human transcription factors: function, expression and evolution. Nat Rev Genet, 10(4):252-63

7: Kummerfeld SK et al., DBD: a transcription factor prediction database. Nucleic Acids Res, 1;34(Database issue):D74-81

8: Wilson D et al., DBD—taxonomically broad transcription factor predictions: new content and functionality. Nucleic Acids Res., 1;36(Database issue):D88-92

9: Hsu, F. et al., The UCSC Known Genes. Bioinformatics, 22(9):1036-1046

10: John S et al., Chromatin accessibility pre-determines glucocorticoid receptor binding patterns. Nat Genet, 43(3):264-268

11: Thurman RE et al., The accessible chromatin landscape of the human genome. Nature, 6;489(7414):75-82

12: Michael I. Love and Wolfgang Huber and Simon Anders, Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. Genome Biology, 15(12):550

13: Plotly Technologies Inc., Collaborative data science, 2015

14: W. Huber, V.J. Carey, R. Gentleman, M. Morgan, Orchestrating high-throughput genomic analysis with Bioconductor. Nature Methods, 12:115-121

15: Lawrence M, Huber W, Pagès H, Aboyoun P, Carlson M, et al., Software for Computing and Annotating Genomic Ranges. PLoS Comput Biol, 9(8):e1003118

16: Steffen Durinck, Paul T. Spellman, Ewan Birney and Wolfgang Huber, Mapping identifiers for the integration of genomic datasets with the R/Bioconductor package biomaRt. Nature Protocols, 4:1184-1191

17: Steffen Durinck, Yves Moreau, Arek Kasprzyk, Sean Davis, Bart De Moor, Alvis Brazma and Wolfgang Huber, BioMart and Bioconductor: a powerful link between biological databases and microarray data analysis. Bioinformatics, 21:3439-3440

18: Marc Carlson and Bioconductor Package Maintainer, TxDb.Hsapiens.UCSC.hg19.knownGene: Annotation package for TxDb object(s).

19: Marc Carlson, org.Hs.eg.db: Genome wide annotation for Human.

20: Hadley Wickham, Romain Francois, Lionel Henry and Kirill Müller, dplyr: A Grammar of Data Manipulation.

21: Yihui Xie, knitr: A General-Purpose Package for Dynamic Report Generation.

22: JJ Allaire et al., rmarkdown: Dynamic Documents for R.

23: H. Pagès, M. Lawrence and P. Aboyoun, S4Vectors: S4 implementation of vectors and lists.

24: Hadley Wickham, scales: Scale Functions for Visualization.