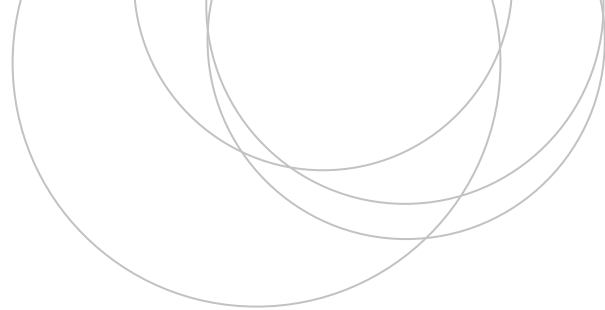




Universidad
del País Vasco

Euskal Herriko
Unibertsitatea

ZIENTZIA
ETA TEKNOLOGIA
FAKULTATEA
FACULTAD
DE CIENCIA
Y TECNOLOGÍA



Bachelor's Thesis

Degree in Biochemistry and Molecular Biology

NaviSE

Superenhancer Navigator integrating epigenomics signal algebra

Author:

Alex Martínez Ascensión

Director:

Beatriz Apéllaniz Unzalu

Tutors:

Ander Izeta Permisán and Marcos Araúzo-Bravo

© 2017, Alex Martínez Ascensión

Leioa, 26 June 2017

CONTENTS

1	Introduction and objectives	1
2	Materials and methods	3
2.1	Preprocessing of NGS files	3
2.2	SE prediction and annotation	4
2.3	Statistics of the comparison between TE and SE	7
2.4	Generation of the NaviSE report	7
2.5	Parallelisation implementation	7
3	Results and discussion	9
3.1	HTML report generation	9
3.2	Process parallelisation	10
3.3	SE prediction of different cell lineages	10
3.3.1	Main page, SE table, and Statistics	10
3.3.2	GOEA and GSEA results	11
3.3.3	Enrichr analysis	13
4	Conclusions	14
5	References	14
6	Appendix I: Supplementary Figures	17
7	Appendix II: Original Article	21
8	Appendix III: NaviSE installation and use manual	41

1 INTRODUCTION AND OBJECTIVES

Enhancers are functionally defined as short (50-1000 bp) regions of DNA which can be bound by activator proteins to increase the transcription of a gene by transcription factors (TFs). Most enhancers are located upstream before the promoter (up to 1 Mpb), although they might be located after a gene. Therefore, enhancers are considered as important regulatory elements of gene expression [Penacchio et al., 2013]. Advances in DNA sequencing technology such as chromatin immunoprecipitation with sequencing (ChIP-seq) have revealed that enhancers contain multiple binding sites for factors such as p300 or presence of ‘activated’ histone marks such as H3K4me1 or H3K27ac. This discovery has broadened the definition of enhancers to include any fragment of DNA with chromatin or factor binding profiles which correlate with a certain function regarding gene expression. Nowadays, a typical ChIP-seq experiment may yield between 10,000 and 150,000 putative enhancers per cell type [Pott et al., 2015].

Superenhancers (SE) are a novel class of transcription regulatory DNA regions with unusually strong enrichment for binding of transcriptional coactivators such as Mediator (MED1), ‘active’ histone marks such as H3K27ac, or cell and tissue-specific TFs, detected by ChIP-seq experiments and determined by the algorithm proposed by Young [Whyte et al., 2013]: (1) Determination of enhancers by ChIP-seq signal peaks (of Med1 binding, H3K27ac presence, etc). Enhancers are inferred peaks previously calculated by a peak finding algorithm, such as MACS (Model-based Analysis of ChIP-seq) [Zhang et al., 2008]. (2) Stitching of enhancers, so that two consecutive enhancers separated by less than an arbitrary length (12,5 kb generally) are combined into the same stitched enhancer. (3) Stitched enhancers are ranked by their signal, and the split point between a SE and a typical enhancer (TE) is defined by the inflexion point in the curve of signal ranking (**Figure 1a**).

As a result of this algorithm, SEs represent large clusters of transcriptional enhancers that drive the expression of ‘master control’ genes that define cell identity. In mouse embryonic stem cells (mESC), for instance, SEs have been shown to regulate genes related to pluripotency, such as master transcription factors like OCT4/POU5F1, SOX2, NANOG, or KLF4; which also bind to their own and to other TF SEs, establishing a transcriptional regulatory circuitry (**Figure 1b** and **1e**).

SEs differ from TEs for having higher TF binding density and number of TF binding sites (TFBSs), which correlate with a much higher expression of their target genes [Whyte et al., 2013] (**Figure 1c** and **1d**). Luciferase report assays have also shown that cloned regions from SEs show higher activity than TEs (**Figure 1f**). Since SEs determine cell differentiation profile [Adam et al., 2015] and specific SEs have been found for each cell type (**Figure 1g** and **1h**), single-nucleotide polymorphisms (SNPs) located in SEs have also been found to be related to genes contributing to diseases such as cancer, Alzheimer’s disease or systemic lupus erythematosus [Hnisz et al., 2013] (**Figure 1i**). Moreover, aberrant SE DNA methylation patterns, as well as particular SE-associated gene sets have been found to be altered in cancer [Heyn et al., 2016, Hnisz et al., 2013] (**Figure 1j**).

Therefore, SEs are genomic elements with high interest in gene transcription regulation. Still, although protocols for SE assessment already exist, there is yet no tool which integrates all the

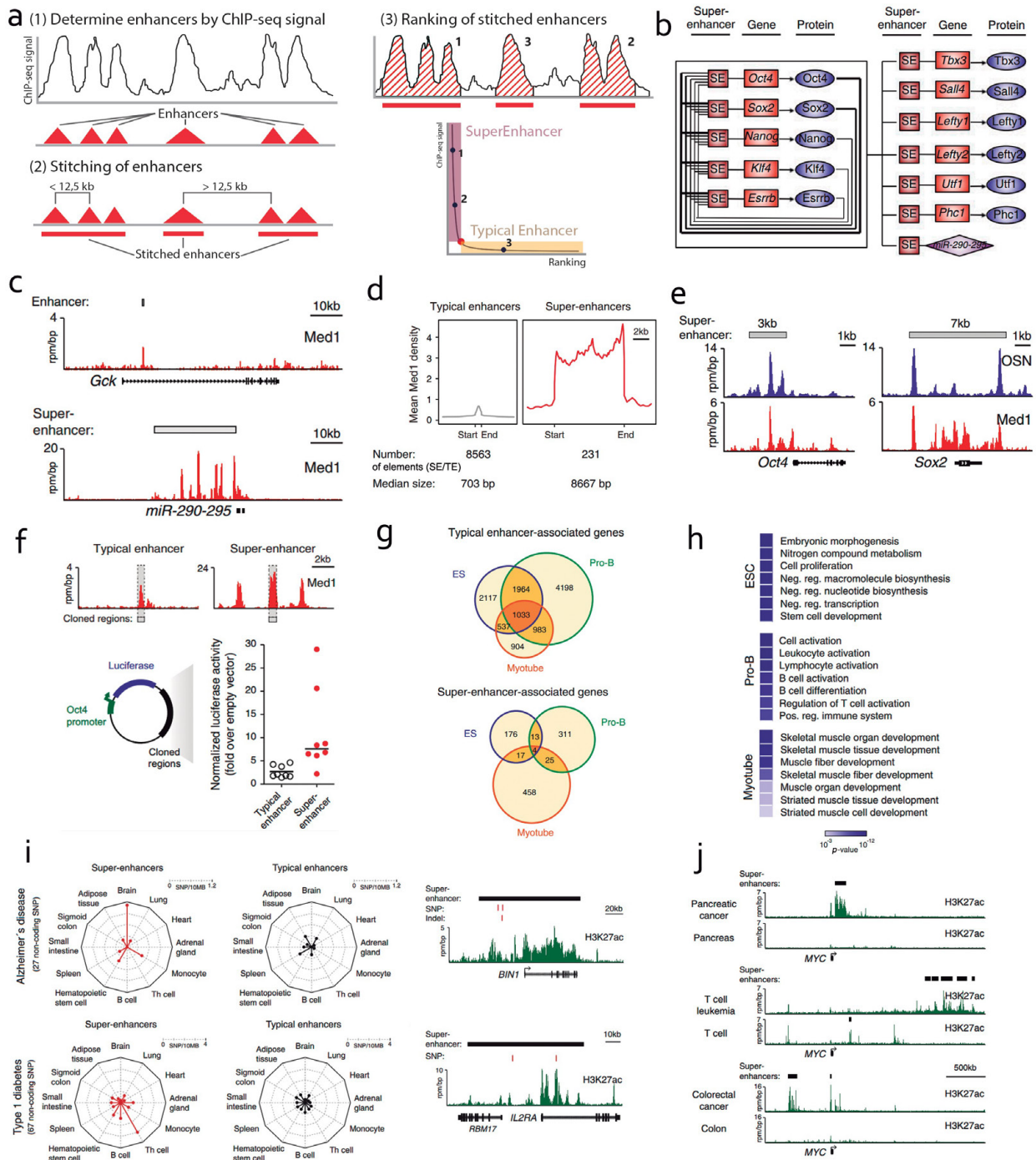


Figure 1. Superenhancer definition and experiments. (a) Scheme of the algorithm for SE prediction. Blue dots in the SE curve indicate hypothetical prediction of two SEs (1 and 2) and a TE (3). The red dot indicates the boundary between TE and SE. (b) Diagram of a portion from the mESC core regulatory circuitry. Master transcription factors (*Oct4*, *Sox2* etc) activate both other genes (*Tbx3*, *Lefty1*, etc) and themselves. (c) ChIP-seq binding profiles of Med1 at a TE related to GSK3 gene and a SE related to miR-290-295 in mESC. Differences in peak signal can be easily discerned. rpm/bp = reads per million/base pair, which indicates the ChIP-seq signal intensity. (d) Normalised ChIP-seq signal across 8563 TEs and 231 SEs, and median of the lengths of SEs and TEs. (e) ChIP-seq binding profiles for the ESC transcription factors *Oct4*, *Sox2*, and *Nanog* (OSN) (blue), and *Med1* (red) to the OCT4 and SOX2 loci in ESC. It can be seen that, according to (b), master transcription factors have high ChIP-seq signal in their own SEs. (f) ChIP-seq binding profiles for Med1 to the SGK1 TE (left) and ESRRB SE (right) loci. Enhancers neighboring selected genes were cloned (gray dashed area) into reporter plasmids containing the Luciferase gene regulated by the Oct4 promoter and were subsequently transfected into ESCs. Differences of reporter assay between SE and TE are significant (p -value = 0.02, n = 8, two tailed t-test). (g) Venn diagrams depicting TE and SE-associated genes in three cell types (ESC, myotube and Pro-B cells). The proportion of shared genes between all cell types is significantly smaller in SEs than in TEs.

Figure 1. (cont.) (h) Top 7 ontology terms associated with SEs in ESC, Pro-B and myotubes. (i) Radar plots showing the density of noncoding SNPs linked to Alzheimer's disease and Type 1 diabetes in the SEs and TEs identified in 12 human cell and tissue types. The center of the plot is 0, and a coloured dot on the respective axis indicates the SNP density (SNP/10 MB sequence) in the SEs or TEs of each cell and tissue type. On the right there is a ChIP-seq binding profile of a SE associated to a gene linked to each disease (BIN1 in Alzheimer's and IL2RA for type 1 diabetes), with red marks depicting SNPs or insertions or deletions (Indels). (j) Significant differences appear in the ChIP-seq binding profiles for H3K27ac surrounding the MYC oncogene in several cancerous tissues (pancreas and colon) and cells (T cells). (b) to (h) figures were obtained and adapted from [Whyte et al., 2013], and (i) and (j) items were obtained and adapted from [Hnisz et al., 2013].

processing stages from the raw data reads from the sequencer, through quality control and reads alignment, to peaks estimation and peaks stitching, ending with a fully-annotated, interactive documentation of the results. Instead, the user has to use each of nearly 20 programs independently, which, in turn, requires a considerable amount of time. Furthermore, CPU resources and running-time are crucial for the high quantity of data produced by Next Generation Sequencing (NGS) technologies, hence another of the main demands in NGS software development is the parallelisation of the most time-consuming processes.

To provide a solution to these demands we have developed NaviSE [Ascensión et al., 2017], a user-friendly tool which automatically processes genome-wide NGS epigenomics data from multiple input files, integrating several epigenomic signals at once with its epigenomics signal algebra, into an interactive HTML report, built with full annotations about SEs, such as associated genes, gene ontology (GO), graphs with metrics and statistical analysis. NaviSE has also been parallelised in the most relevant and time-consuming processes, in order for the user to run multiple analysis in a significantly reduced amount of time. Finally, NaviSE is developed for users with working knowledge in informatics (notions of Unix systems and Python programming language).

2 MATERIALS AND METHODS

2.1 PREPROCESSING OF NGS FILES

Before the determination of SEs, NaviSE prepares the raw data, allowing multiple replicates and controls at once. The main steps for such preprocessing are as follows:

1. *Input format file recognition and file processing*: NaviSE recognizes multiple file formats, e.g., .sra, .fastq, .sam, .bam and .bed, and transforms an *upstream* format (.sra, .fastq, .sam) into a .bam file.
2. *Alignments*: Performed by default with bowtie2 [Langmead and Salzberg, 2012]; then, alignment.sam files are processed to .bam files by samtools. NaviSE also allows read alignment with MOSAIK [Lee et al., 2014], STAR [Dobin et al., 2013] and BWA [Li and Durbin, 2009] aligners. Users may also generate their own .sam or .bam files with other aligners, and NaviSE will recognize them for further processing.
3. *Quality control with FastQC*: NaviSE performs the quality analysis of the reads from the .fastq files using FastQC, analysing per base quality, GC content, *k*-mers, etc.
4. *Combination of replicates and peak calling with MACS*: NaviSE calculates signal peaks with

MACS (Model-based Analysis for ChIP-Seq)[Zhang et al., 2008].

2.2 SE PREDICTION AND ANNOTATION

Once the data is preprocessed, a SE prediction and ranking is performed. SEs then are further analysed in search of SE related genes, DNA sequence motifs, GO terms or statistical estimations.

1. *Epigenomics signal algebra*: In case more than one epigenomic signal was used to predict SEs, NaviSE integrates all the signals to improve the SE prediction. Different epigenomic signals are defined by the names of the signal data files $Sig \in \{H3K27ac, H3K4me1, ATAC-seq, \dots\}$, and operators are defined by $Ope \in \{AND, OR, NOT, XOR, +, - SYM\}$. The way these algebra operators have been adapted is illustrated in **Figure 2**. When performing ‘epigenomics signal algebra’, NaviSE picks the first pair of signals separated by each operator, starting from the left, and the results from the operation are combined with the next signal using the next operator, until the last signal identifier is reached.

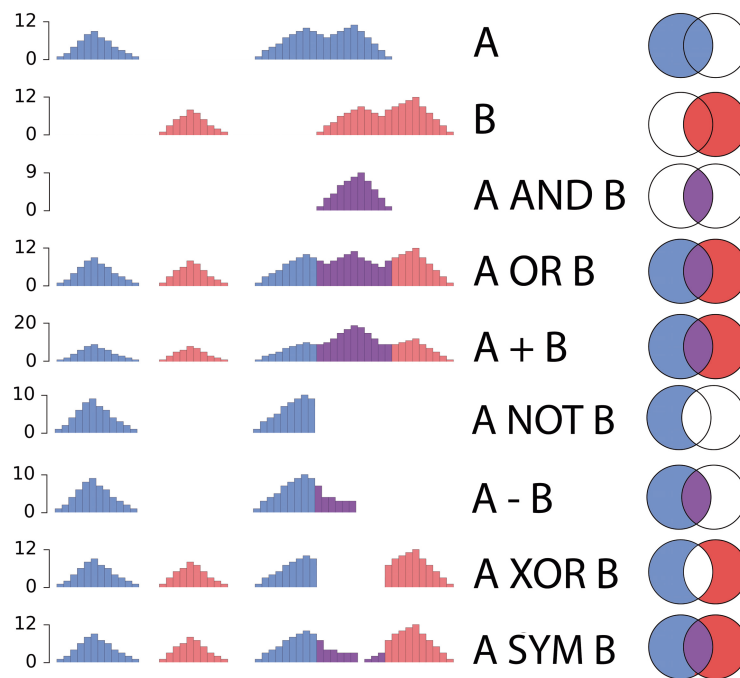


Figure 2. Epigenomics signal algebra. Schemes of the different pairwise operations implemented in NaviSE. The top two rows depict an example of the two epigenomic signals to be combined, and the remaining rows illustrate the signal profile after applying the respective operator. Simplified Euler-Venn diagrams given in the rightmost column illustrate the set operations. AND, OR, NOT and XOR are Boolean operations which do not change the signal pileup; whereas +, -, and SYM are arithmetic operations which can change the signal pileup. In the case of - and SYM, negative pileups are transformed into zeros.

2. *SE prediction*: To predict the SEs in a sample, NaviSE performs the *stitching* of MACS peaks which fall within a threshold distance, using our own implementation of the algorithm developed by Young’s lab [Whyte et al., 2013], previously described in the introduction.
3. *SE gene assignment*: Once the SE locations are determined, each SE is assigned a gene by proximity with the closest transcription start site (TSS). NaviSE also includes information about genes overlapping the SE or genes proximal to each SE.

4. *Subpeak annotation*: The SEs and TEs subpeaks have been shown to act synergistically within the SE despite being individual and independent structures [Hnisz et al., 2015]. To provide information about the SE subpeaks structure and location, NaviSE performs an annotation of the subpeaks that represent each SE. The annotation contains the following parameters:
 - Number of subpeaks, *loci* and TSS locations.
 - Association to TSSs: To understand the regulatory role of the SEs, it is important to resolve their association to TSSs. This analysis portrays the percentage of subpeaks outside the range of a user-defined distance within the TSS and a classification of the SEs according that percentage: *Pure* if all the subpeaks are outside the TSS, *Only TSS* if all the subpeaks lay within the TSS, and *Mixed* if there are both types of subpeaks.
5. *Automatic generation of SE peak distribution profiles*: To visualize the SE peak distribution we have implemented our own Genome Viewer Tool (GVT). With this tool two snapshots at *near* and *far* distances for each SE are portrayed. With *near* shot the user is able to determine the morphology of the SE and with *far* the user is able to locate the SE in its genomics surroundings.
6. *HOMER motif finding*: SEs enclose a high number of TFBSs [Whyte et al., 2013], therefore identifying such TFBSs is important for SE annotation. To find motifs of TFBSs that are specifically enriched in the *loci* of SEs, relative to the *loci* of TEs, NaviSE uses the Hypergeometric Optimization of Motif Enrichment (HOMER). After the analysis, a *HOMER table* is generated, which includes motifs enriched in SEs, as well as list of *de novo* motifs predicted by HOMER.
7. *Gene Ontology Enrichment Analysis (GOEA)*: To predict the functionality of the SEs, based on the closest gene of each SE, determined by HOMER, NaviSE uses goatools [Haibao et al.].
8. *Pathways and protein-protein interaction annotation*: To obtain annotation of TFs and pathways related to SEs, NaviSE uses Enrichr [Chen et al., 2013]. To obtain protein-protein interaction (PPI) networks of SEs, NaviSE uses the database of PPIs String [Szklarczyk et al., 2015].
9. *NaviSE GUI*: To navigate throughout all the results, we have implemented an interactive chromosomal plot (**Figure 3**) that represents the SE location in a karyotype; alongside with graphs that depict statistical values and properties related to SEs.

Chromosomal plots are designed to include *hot-spots* with links to other elements from the final report, which are activated when the user navigates with the mouse over the gene names on the chromosomal plot. NaviSE generates three types of chromosomal plots:

- Enrichment plot: it shows the *loci* location and the chromosome enrichment or depletion.

- Rank plot: it shows *loci* coloured according to their rank. Several percentiles are represented based on the rank of the SE, and SEs falling within a percentile will be coloured with their corresponding colour.
- Closeness plot: it represents the proximity between SEs, according to which SEs will be coloured. This plot is highly useful to discern clusters of SEs that look overlapped. For the ordered list $\{SE_1, SE_2, \dots, SE_{a-1}, SE_a, SE_{a+1}, \dots, SE_{c-1}, SE_c\}$, of c SEs within a chromosome, for which each SE_k support is defined by its start ($SE_{k,start}$) and end ($SE_{k,end}$) *loci* positions, its distance $C(SE_k)$ to the closest SE is determined by the following expression:

$$C(SE_k) = \begin{cases} SE_{2,start} - SE_{1,end} & \text{if } k = 1 \\ \min(SE_{a+1,start} - SE_{a,end}, SE_{a,start} - SE_{a-1,end}) & \text{if } k = a \\ SE_{c,start} - SE_{c-1,end} & \text{if } k = c \end{cases} \quad (1)$$

In all these chromosomal plots a probability p determining whether a chromosome is enriched (marked with \wedge for $p < 0.05$ and $\wedge\wedge$ for $p < 0.01$) or depleted (marked with Υ for $p < 0.05$ and $\Upsilon\Upsilon$ for $p < 0.01$) with SEs is calculated by a binomial approximation of the hypergeometric distribution ($h(k; K, n, N) \rightarrow b(k; K, z)$; $z = \frac{n}{N}$) [Jaioun and Teerapabolarn, 2014], where N is the number of genes in the whole genome, K is the number of SEs in all chromosomes, n is the number of genes in that chromosome, and k is the number of SEs in that chromosome.

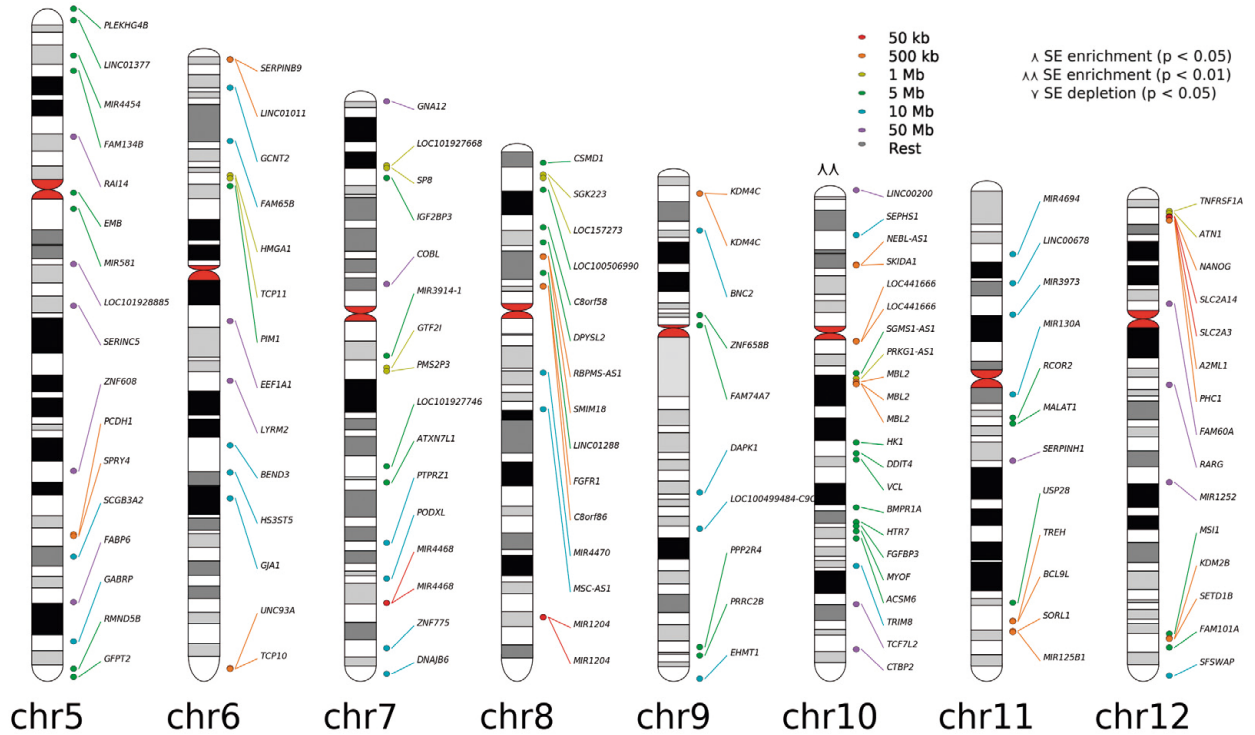


Figure 3. Chromosomal plot. Partial snapshot of the chromosomal plot of SE for ESC and H3K27ac histone mark. Hot-spots and line colours represent distances between two SEs (red to violet represents smaller to bigger distances), and \wedge / $\wedge\wedge$ and Υ / $\Upsilon\Upsilon$ represent chromosome enrichment or depletion in genes with statistical significance p of 0.05 and 0.01, respectively.

10. *Gene Set Enrichment Analysis (GSEA)*: To obtain additional functional annotation of SEs, NaviSE performs the GSEA [Subramanian et al., 2005] from SE-associated genes, using gene sets from the Molecular Signatures Database (MSigDB).

2.3 STATISTICS OF THE COMPARISON BETWEEN TE AND SE

Although both SEs and TEs derive from MACS peaks, they structurally differ for having higher peak density. To illustrate the differences between SEs and TEs, NaviSE shows in the final report a collection of plots depicting the differences between them.

1. Ranking of SEs by the order of SE score.
2. INs and OUTs: It shows statistics about the percentages of SEs and TEs that lay within a TSS or not. This might be interesting if a sample contains an elevated percentage of SEs within TSS, as some of these SEs might be misinterpreted as promoter signals.
3. SE vs TE length distribution: It shows the distribution of SE and TE length and pileup in a double histogram and a scatter plot. The histogram lying on the X axis of the scatter corresponds to the length of SEs and TEs; and the histogram on the Y-axis corresponds to the pileup.
4. SE vs TE subpeak length distribution: Similar to the SE vs TE length distribution graph, although showing the distribution of enhancers inferred by MACS.
5. Number of subpeaks (for number of bins $N = 10$ and $N = 20$) of SEs and TEs.

2.4 GENERATION OF THE NAVISE REPORT

The final step of NaviSE is the generation of an HTML report with several windows, each of which contains interactive links to external websites providing further information about SEs, and internal pages created by NaviSE within the report. The content of this report is discussed in detail in *Results* section.

2.5 PARALLELISATION IMPLEMENTATION

The algorithm of parallelisation developed in NaviSE constitutes a significant improvement of performance in the analysis of NGS samples compared to non-parallelised pipelines. For the parallelisation, NaviSE determines the optimal number of processes, k , compatible with the computer resources as Luu *et al.* do in [Luu et al., 2017]. Such resources are the parallel processing capability of the computer measured as the number of cores, C , and the total main memory, M , in Gigabytes GBs. NaviSE optimizes automatically, for each processing task i , the number of threads, k_i in which the task i will be parallelised by the expression:

$$k_i = \min(C, C_u, \lfloor M/m_i \rfloor, l_i) \quad (2)$$

where C_u is the maximum number of cores reserved by the user to run NaviSE, m_i is the main computer memory, measured in GBs, needed to run one process in task i , $\lfloor \cdot \rfloor$ is the floor operator and

l_i is the cardinal of $D_i = \{d_1, d_2, \dots, d_m\}$ which is the set of *chunks* of distributed data elements to be processed in task i . If $l_i > k_i$, the first k_i chunks are distributed to k_i threads. The distribution of information (number of chromosomes for stitching, SE peak distribution profiles for GVT, number of gene sets for GSEA) to be parallelised is based on a cyclic algorithm, implemented in Python with the following outline: For the ordered set $S_i = \{s_1, s_2, \dots, s_n\}$ of information elements, the set $P_i = \{1, \dots, k_i\}$ of processes and for the set D_i of data (chromosomes, positions on a list, gene sets) to be distributed across processors, we define D_{pi} as the *chunk* of data of the task i that is assigned to each processor p :

$$D_{pi} = \{d_j \mid \forall d \in D_i, p \in P_i, j \in \{1, \dots, l_i\}, j \bmod k_i = p\} \quad (3)$$

where mod is the module operator. Once the *chunk* D_{pi} is constructed, the subset of information elements $S_{D_{pi}} \subset S_i$ will be defined depending on the task i which is being parallelised. The list of parallelised tasks in NaviSE is $i = \{\text{STIT (stitching of SEs), GVT (taking snapshots of SEs), GSEA, HOMER}\}$. An example for STIT parallelization, for better understanding of the process, is developed in **Figure 4**.

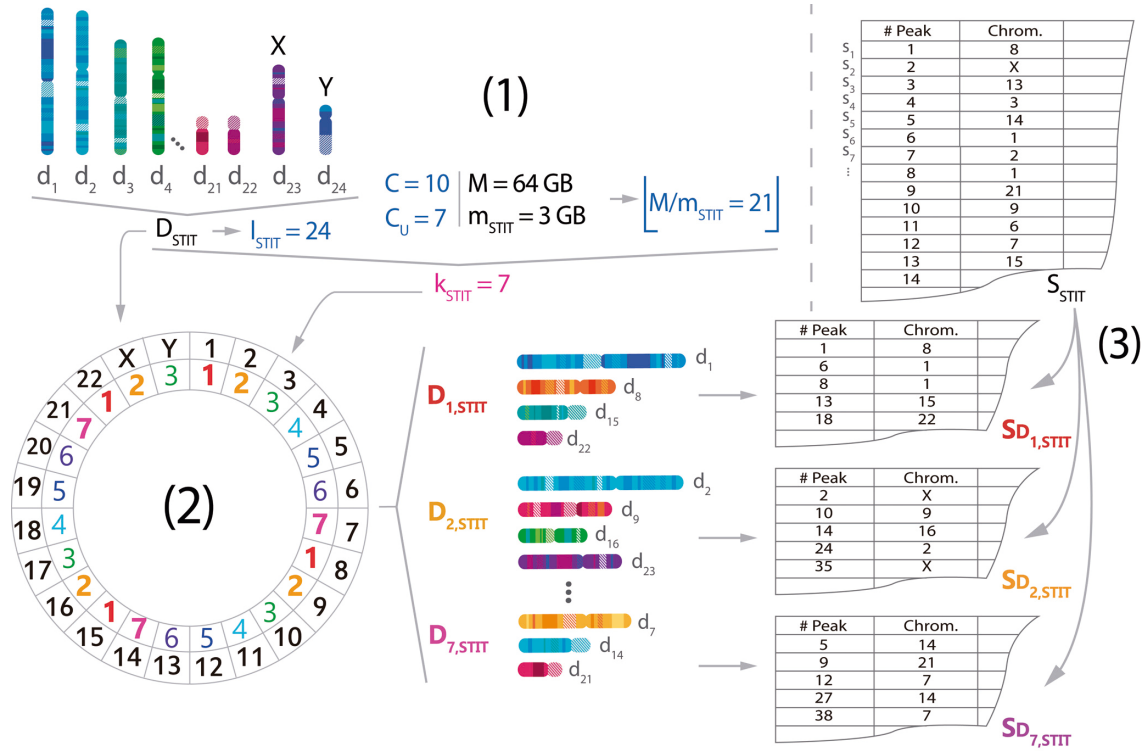


Figure 4. Scheme of parallelization of stitching. (1) Determination of the number of processes (k_{STIT}) based on **Equation 2** for a case in which the number of cores (C) is 10, the maximum number of cores reserved by the user (C_u) is 7, the memory of the computer (M) is 64 GB, the memory allocated to stitching (m_{STIT}) is 3 GB and the cardinal (l_{STIT}) of the set of chromosomes ($D_{\text{STIT}} = \{d_1 = 1, d_2 = 2, \dots, d_{22} = 22, d_{23} = X, d_{24} = Y\}$) is 24. The resulting number of allocated cores calculated by **Equation 2** is $k_{\text{STIT}} = C_u = 7$. (2) Construction of data *chunks* is calculated by **Equation 3**. Since $k_{\text{STIT}} = 7$, the set of chromosomes, D_{STIT} , is divided into 7 subsets or *chunks*: $D_{1,\text{STIT}} = \{d_1, d_8, d_{15}, d_{22}\}$; $D_{2,\text{STIT}} = \{d_2, d_9, d_{16}, d_{23}\}$; \dots ; $D_{6,\text{STIT}} = \{d_6, d_{13}, d_{20}\}$ and $D_{7,\text{STIT}} = \{d_7, d_{14}, d_{21}\}$. (3) Assignment of information elements. In the case of stitching, assigned elements are MACS peaks (inferred as enhancers). After the assignment of the subsets $D_{1,\text{STIT}}$, $D_{2,\text{STIT}}$, etc., the set of MACS peaks, $S_{\text{STIT}} = \{s_1, s_2, \dots\}$ is divided into 7 subsets of elements, $S_{D_{1,\text{STIT}}} = \{s_1, s_6, s_8, \dots\}$, $S_{D_{2,\text{STIT}}} = \{s_2, s_{10}, s_{14}, \dots\}$, \dots , $S_{D_{7,\text{STIT}}} = \{s_5, s_9, s_{12}, \dots\}$, based on the chromosome of each row. Finally, all the subsets of elements are simultaneously processed by NaviSE, combined into one file, and the SE ranks are calculated.

3 RESULTS AND DISCUSSION

To illustrate the performance of NaviSE, we have selected H3K27ac histone mark, downloaded from the GEO database [Edgar et al., 2002] for three cell types: human Embryonic Stem Cells (ESC) (GSM663427, with control GSM605335), monocytes (MON) (GSM1003559 with control GSM1003475) and neurons (NEU) (GSM2072642, with control GSM2072639). We have also used H3K4me1 (GSM409307) and H3K4me3 (GSM409308) from ESCs.

3.1 HTML REPORT GENERATION

The output of NaviSE is a collection of HTML linked pages containing a blue horizontal ribbon with links to all the HTML pages from the report, detailed below; a grey sidebar by which the user can access the different subsections; and a window in which the results are displayed.

The *main window* contains basic information about the analysis and different chromosomal plots, defined in the point 9 of *SE prediction and annotation* section, represented in the chromosomal plot snapshot of **Figure 3**. The chromosomal plot includes links to the SEs in *SE Table* window, which includes general information about each SE (genomic location, number of subpeaks, SE score), alongside with a snapshot of the SE genomic signal profile, included for visual evaluation of the SE quality, together with the quantitative SE score. The *SE Table* columns referring to gene names and genomic location include, respectively, a link to GeneCards site [Rebhan et al., 1997] and UCSC Genome Browser [Tyner et al., 2017], as shown in **Figure S1**.

Statistics window implements graphs depicted in *Statistics of the comparison between TE and SE*. Some of those graphs are analysed thoroughly in the corresponding *Analysis of different cell lineages* results section.

GOEA window includes the results from the GOEA. A barplot shows the significant terms from GO categories (biological process, cellular component and molecular function) which, upon clicking, will lead to graph of the GO terms associated with the significant term, each of which contains the related genes associated to that term. Below the barplot, there is a table which includes values such as enrichment ratio of the predicted cell population, and the False Discovery Rate (FDR) for each term.

Similarly, the *GSEA* window (**Figure S2**) contains several graphs depicting the GSEA profile for each signature (group of gene sets) and threshold. Clicking on a graph leads to its corresponding element on a table below, with the significant GSEA term, related SE genes, and statistical values linked to the GSEA term such as Enrichment Score (ES), Normalized Enrichment Score (NES), FDR and *p*-values provided by GSEA.

HOMER window shows the results from the motif analysis by HOMER, which includes two ranked tables, one for known motifs and another one for *de novo* motifs. ‘Known motifs’ table contains a LOGO image for each motif and the name of the TF or binding protein using such binding motif. It also includes the percentage of SE and TE sequences that such motif has, and a *p*-value of the association of the SE with such motif. The *de novo* table includes motifs predicted by HOMER to bind elements differentially in SEs and TEs. Upon clicking on each element in *de novo* table,

NaviSE redirects to a HOMER-generated page that includes more information about the motif.

Finally, *StringDB* and *Enrichr* windows show, respectively, PPI networks from SEs; and results from Enrichr website including TFs related to SEs, cell or tissue specification or metabolic pathways linked to the SE population. Each subsection includes a barplot of the significant terms which link to the elements in a specific table.

3.2 PROCESS PARALLELISATION

The parallelisation of NaviSE is fundamental to save time during the data processing, more so when the analysis is performed simultaneously with numerous cell types or marks. The computing time optimization achieved upon NaviSE parallelisation is shown in **Figure 5**.

Most time-consuming processes show a considerable decrease in running time: in SE prediction up to a 30 % of the original time, in gene annotation up to 10 %, and GVT up to 8.5 %. In short, the overall amount of time is reduced up to a 40 % between 1 and 19 processors, and the optimal difference is achieved at 15 processors, with a reduction up to 30 %. Hence, NaviSE shows a considerable reduction of processing time even with small processing capability, below 6 CPUs, which may allow conducting research with mid-range computers.

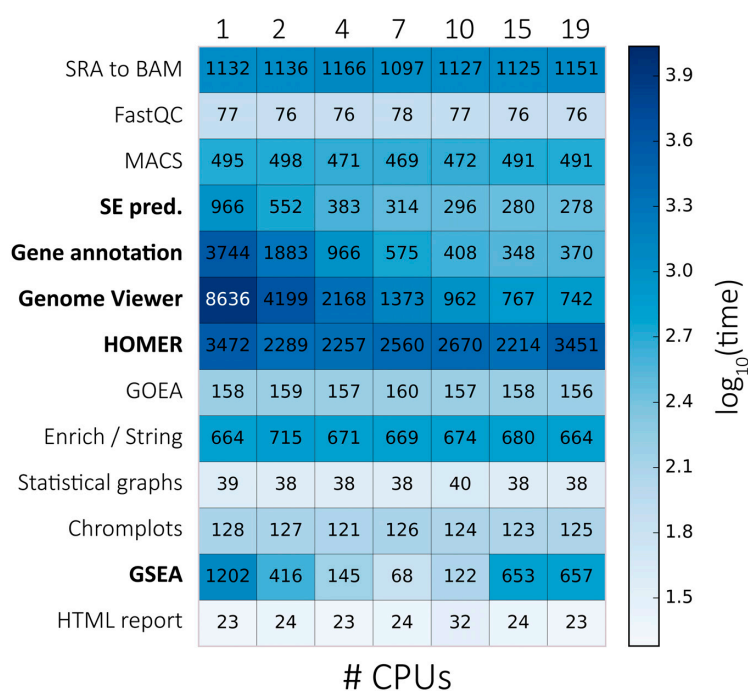


Figure 5. NaviSE performance comparisons. NaviSE CPU running time for different numbers of CPUs. Heatmap of the processing time for each NaviSE process for different numbers of CPUs, written on top. Tasks parallelised by NaviSE are highlighted in bold typeface.

3.3 SE PREDICTION OF DIFFERENT CELL LINEAGES

3.3.1 Main page, SE table, and Statistics

Using the same default parameters with H3K27ac histone mark, NaviSE analysis for the different cell lines yielded a wide range of SEs ($n_{ESC} : 664$, $n_{NEU} : 1073$, $n_{MON} : 1235$). The signals of the most important SEs are shown in the **Figure 6** and the main statistics for each cell type are depicted

in the **Figure 7**.

The distribution of subpeaks varies considerably between SEs and TEs. TE subpeak distribution follows a Zipfian-like distribution in all the analysed cell lines; whereas the SE distribution might follow a χ^2 distribution or a normal distribution. In the case of ESCs, the maximum of subpeaks is between 5 and 7, in NEU and MON the distribution is uniform between 6 and 14 subpeaks, with a considerable amount of SEs having more than 20 subpeaks.

The differences in length distribution between TEs and SEs are apparent in all samples. Interestingly, TEs usually show a bi or trimodal distribution with maxima at ~ 100 , ~ 1000 or ~ 10000 nt in all the analysed cell types, whereas SEs show a monomodal normal-like distribution with means around 25000 - 50000 nt. On the other hand, subpeak distribution shows no significant differences between SEs and TEs, both in length and pileup.

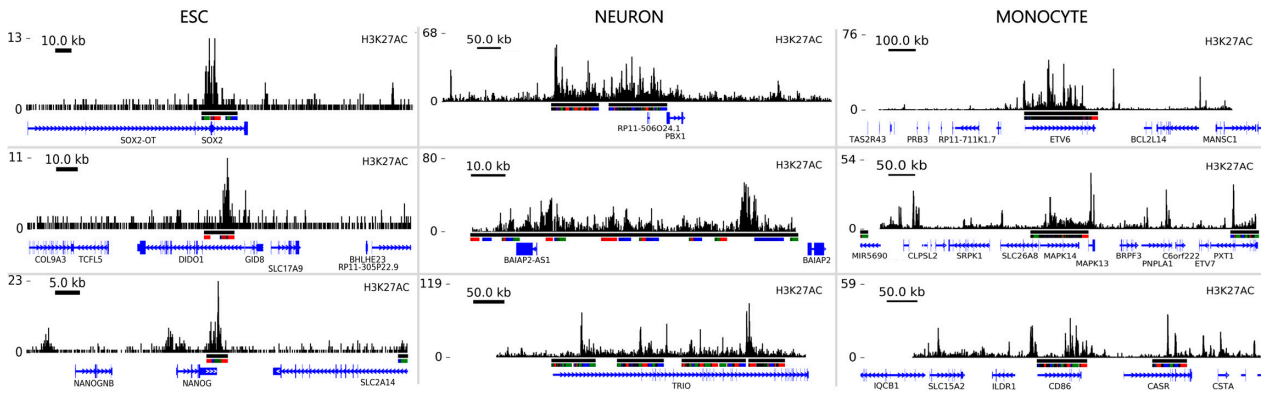


Figure 6. SE ChIP-seq peak distribution. Box plot of peak distribution for each SE obtained with our GVT module. Each column represents SEs from each cell type. Black lines below the signal represent the SE supports at each sample, and bars in alternating colours below SE bar show the supports of the SE subpeak composition.

3.3.2 GOEA and GSEA results

GOEA and GSEA are represented in **Figure S3** and **Figure S4** respectively. Both analysis show correlation of functions to each cell type.

For ESC, the most relevant GO terms are related to protein expression (*positive regulation of transcription*), rearrangement of cellular morphology (*focal adhesion, lamellipodium*) or pluripotency (*somatic stem cell population maintenance*). As for GSEA, significant terms are related to master TFs of ESCs, such as *NANOG*, or cytoskeletal reorganization. Among the predominant genes, we remark *ROR1* (that modulates neurite growth and is highly expressed during early embryonic development [Azfal and Jeffery, 2003]), *ZIC3/5* (involved in the formation of right/left axis during development, and direct activator of *NANOG* promoter in ESC [Lim et al., 2010]) or *SOX2* (one of the Yamanaka reprogramming TFs, used for the induction of pluripotency, as well as a core pluripotency factor in ESC [Takahashi and Yamanaka, 2006]).

Regarding NEU, the most relevant GO terms are related to neural development (*ephrin signaling, Wnt signaling pathway, dendritic spine, axon guidance*). As for GSEA, three relevant terms are *generation of neurons, neuron differentiation, and neurite development*, whose related genes are *CDK5R1* (neuron-specific activator of cyclin-dependent kinase 5, required for proper development

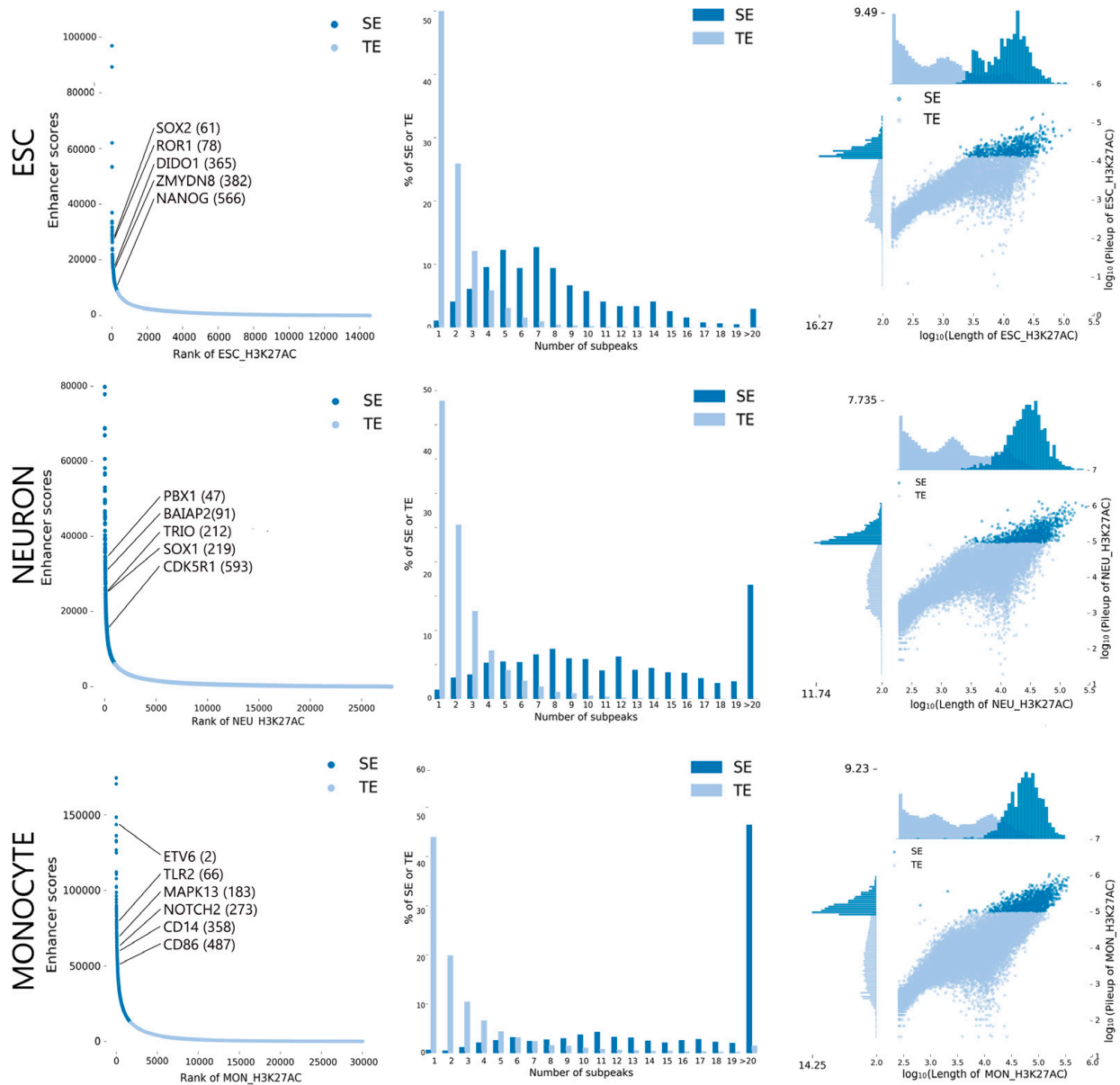


Figure 7. NaviSE GUI statistics. Diagrams of statistical parameters for each cell type, arranged in rows. SE ranking by ChIP-seq signal (left column) with the most relevant SEs of each cell type and their corresponding ranking, SE distribution of the number of subpeaks of the SEs (center column) and SE pileup vs length scatter plots in log₁₀ scale with the respective distributions of SE pileups in ordinates and SE lengths in abscissas (right column).

of the central nervous system, also found essential for oligodendrocyte maturation and myelination [Luo et al., 2016]), *BAIAP2* (brain-specific angiogenesis inhibitor binding protein, might be related to neural growth-cone guidance, dendritic spine development and NMDA receptor regulation [Kang et al., 2016]) and *PBX1* (regulates differentiation and survival of certain neurons, and is impaired in Parkinson's disease [Castro, 2016]).

Regarding MON, the most relevant GO terms are related to specific functions of monocytes involved with immune response (*phagocytosis*, *T cell receptor signaling pathway*, *lipopolysaccharide-mediated signaling pathway*). As for GSEA, three relevant terms are *T cell receptor signaling pathway*, *reactome immune system* and *immune system process*. Genes shared by several GO terms are

NOTCH2 (related to hematopoiesis), *CD14* (one of the main markers of monocytes), *TLR2* (Toll-like receptor 2, which plays a fundamental role in pathogen recognition and activation of innate immunity), *MAPK13* (is activated by proinflammatory cytokines and cellular stress [Hu et al., 1999]) or *LYN* (might be involved in the regulation of mast cell degranulation, and erythroid differentiation [Parravicini et al., 2002]).

3.3.3 Enrichr analysis

We performed an Enrichr analysis in order to search genes involved in cellular processes related to each cell type. Most of the found genes, if not mentioned previously, appeared in GSEA and GOEA as well.

For ESC, the Enrichr Reactome presents several terms such as *transcriptional regulation of pluripotent stem cells*; and *POU5F1, SOX2, NANOG genes related to proliferation*, widely related to embryogenesis. Predominant genes are *FGF2* (implicated in a multitude of physiologic and pathologic processes, including limb development and tumour growth [Ortega et al., 1998]), *SOX2* or *NANOG* (TF belonging to Homeobox proteins, critically involved with self-renewal of undifferentiated ESCs, which is also one of the Thompson's reprogramming factors [Yu et al., 2007]). ENCODE and Chromatin Enrichment Analysis (ChEA) TFs include TFs related to pluripotency (*TCF3, NANOG, SOX2, POU5F1* and *KLF4* as the most relevant) which share several genes, such as *ZMYDN8*, or *DIDO1* (involved in apoptosis, autophagy, and meiosis). Interestingly, and as described by [Hnisz et al., 2013], we found that SEs predicted by NaviSE are capable of disclosing a crosstalk between TFs (for instance, all the aforementioned TFs interact with *SOX2* and *NANOG*, according to ENCODE).

As for NEU, Reactome includes significant terms such as *axon guidance or semaphorin interactions*, with genes such as *TRIO* or *CDK5R1*; which also appear as genes associated with several TFs such as *REST* (transcriptional repressor that represses neuron-specific genes, such as type II sodium channel gene [Jayhong et al., 1995]), determined by ENCODE or TRANSFAC. A gene predicted to associate with *REST* is *SOX1*, a known neuronal marker.

Regarding MON, Reactome presents several terms such as *immune system, innate immune system, hemostasis or toll-like receptor 2 cascade*, widely related to monocytes, whose associated genes are *TLR2* (plays a fundamental role in pathogen recognition and activation of innate immunity [Jin et al., 2007]), *FOS* (implicated as regulator of cell proliferation, differentiation, and transformation, associated with B lymphocyte differentiation and involved in lipopolisaccharide and low density lipoprotein response [Kang et al., 2010]) or *CD86*, expressed by antigen-presenting cells. Binding of this protein to CD28 antigen is a co-stimulatory signal for activation of the T-cell. TRANSFAC and ENCODE include genes associated with TFs like *GATA1, GATA2, SPI1* or *RUNX1*, among which there are *IKZF1* or *JARID2*. Enrichr also determined markers for monocytes or lymphoid cells, such as *RIN3, CXCR4, TREM1* or *ETV6*.

4 CONCLUSIONS

We designed NaviSE to perform a parallelised SE prediction from genome-wide epigenetic signals, or an algebra of them, providing a comprehensive annotation of SEs. NaviSE SE annotation runs from the motifs of TFBSs enriched in SEs through functional analysis (GOEA, GSEA and enriched metabolic pathways) to PPI networks to a broad tissue prediction, thus, covering a wide range of valuable information for research; of paramount importance due to the regulatory nature of the SEs, which have been described as key players in the determination of cell fate and in the involvement in the mechanisms of disease.

Furthermore, the automatic recognition of multiple file formats and the capability of working with replicates and controls, alongside with the possibility of integrating onto other pipelines or running multiple samples with multiple replicates and signal algebras at once with a simple script in Python, makes NaviSE a foremost tool for an efficient study of SEs. Due to all the capabilities, NaviSE is a time-saving and user-friendly tool for SE analysis.

To validate the biological performance of NaviSE, we applied it to predict the SEs on real data sets of several cell types with a different level of differentiation and commitment, and predicted in all cases SE-associated genes in agreement with the expected cell-specific markers. Thus, in the case of ESCs NaviSE predicted SEs on the ESC markers *NANOG* and *SOX2*, in the case of neurons it predicted the *SOX1* and *CDK5RI* neuron markers, and in the case of monocytes, predicted the *CD86* and *CXCR4* monocyte markers.

Appendixes provide four supplementary figures, the original article published in *BMC Bioinformatics*, and a complete guide to the software installation and use instructions.

5 REFERENCES

- RC Adam, H Yang, et al. Pioneer factors govern super-enhancer dynamics in stem cell plasticity and lineage choice. *Nature*, 521(7552):366–370, 2015.
- AM Ascensión, M Arrospide-Elgarresta, et al. NaviSE: superenhancer navigator integrating epigenomics signal algebra. *BMC Bioinformatics*, 18(296):1–18, 2017.
- AR Azfal and S. Jeffery. One gene, two phenotypes: ROR2 mutations in autosomal recessive Robinow syndrome and autosomal dominant brachydactyly type B. *Human Mutation*, 22(1):1–11, 2003.
- DS Castro. One more factor joins the plot: Pbx1 regulates differentiation and survival of midbrain dopaminergic neurons. *EMBO Journal*, 35(18):1957–1959, 2016.
- EY Chen, CM Tan, et al. Enrichr: interactive and collaborative HTML5 gene list enrichment analysis tool. *BMC Bioinformatics*, 14(128):1–14, 2013.
- A Dobin, CA Davis, et al. STAR: ultrafast universal RNA-seq aligner. *Bioinformatics*, 29(1):15–21, 2013.
- R Edgar, M Domrachev, and AE Lash. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Research*, 30(1):207–210, 2002.
- T Haibao et al. GOATOOLS: Tools for Gene Ontology. *Zenodo*. 10.5281/zenodo.31628.
- H Heyn, E Vidal, et al. Epigenomic analysis detects aberrant super-enhancer DNA methylation in human cancer. *Genome Biology*, 17(11):1–16, 2016.
- D Hnisz, BJ Abraham, et al. Super-Enhancers in the Control of Cell Identity and Disease. *Cell.*, 155:934–947, 2013.
- D Hnisz, J Schuijers, et al. Convergence of developmental and oncogenic signaling pathways at transcriptional super-enhancers. *Molecular Cell*, 58:1–9, 2015.

- MC Hu, YP Wang, et al. Murine p38-delta mitogen-activated protein kinase, a developmentally regulated protein kinase that is activated by stress and proinflammatory cytokines. *Journal of Biological Chemistry*, 274(11):7095–7102, 1999.
- K Jaïoun and K Teerapabolarn. An improved binomial approximation for the hypergeometric distribution. *Applied Mathematical Sciences*, 8(13):613–617, 2014.
- AC Jayhong, J Tapia-Ramírez, et al. REST: A Mammalian Silencer Protein That Restricts Sodium Channel Gene Expression to Neurons. *Cell*, 80:949–957, 1995.
- MS Jin, SE Kim, et al. Crystal structure of the TLR1-TLR2 heterodimer induced by binding of a tri-acylated lipopeptide. *Cell*, 130:1071–1082, 2007.
- J Kang, H Park, and Kim E. IRSp53/BAIAP2 in dendritic spine development, NMDA receptor regulation, and psychiatric disorders. *Neuropharmacology*, 100:27–39, 2016.
- JG Kang, HJ Sung, et al. FOS expression in blood as a LDL-independent marker of statin treatment. *Atherosclerosis*, 212(2):567–570, 2010.
- B Langmead and S Salzberg. Fast gapped-read alignment with Bowtie 2. *Nature Methods*, 9:357–359, 2012.
- WP Lee, MP Stromberg, et al. MOSAIK: A Hash-Based Algorithm for Accurate Next-Generation Sequencing Short-Read Mapping. *PLoS One*, 9(3):1–11, 2014.
- H Li and R. Durbin. Fast and accurate short read alignment with Burrows-Wheeler Transform. *Bioinformatics*, 25:1754–1760, 2009.
- LS Lim, FH Hong, et al. The pluripotency regulator *Zic3* is a direct activator of the *Nanog* promoter in ESCs. *Stem Cells*, 28(11):1961–1969, 2010.
- F Luo, J Zhang, et al. The Activators of Cyclin-Dependent Kinase 5 p35 and p39 Are Essential for Oligodendrocyte Maturation, Process Formation, and Myelination. *Journal of Neuroscience*, 36(10):3024–3037, 2016.
- PL Luu, D Gerovska, et al. P3BSSEQ: Parallel processing pipeline software for automatic analysis of bisulfite sequencing data. *Bioinformatics*, 33(3):428–431, 2017.
- S Ortega, M Ittmann, , et al. Neuronal defects and delayed wound healing in mice lacking fibroblast growth factor 2. *PNAS*, 95(10):5672–5677, 1998.
- V Parravicini, M Gadina, et al. Fyn kinase initiates complementary signals required for IgE-dependent mast cell degranulation. *Nature Immunology*, 3(8):741–748, 2002.
- LA Penacchio, W Bickmore, et al. Enhancers: five essential questions. *Nature Review Genetics*, 14:288–295, 2013.
- S Pott, JD Lieb, et al. What are super-enhancers. *Nature Genetics*, 193(5):8–12, 2015.
- M Rebhan, V Chalifa-Caspi, et al. GeneCards: integrating information about genes, proteins and diseases. *Trends in Genetics*, 13(4):163, 1997.
- A Subramanian, P Tamayo, et al. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *PNAS*, 102(43):15545–15550, 2005.
- D Szklarczyk, A Franceschini, et al. STRING v10: protein-protein interaction networks, integrated over the tree of life. *Nucleic Acids Research*, 43:D447–452, 2015.
- K Takahashi and S. Yamanaka. Induction of Pluripotent Stem Cells from Mouse Embryonic and Adult Fibroblast Cultures by Defined Factors. *Cell*, 126(4):663–676, 2006.
- C Tyner, GP Barber, et al. The UCSC Genome Browser database: 2017 update. *Nucleic Acids Research*, 4(45):D626–D634, 2017.
- A Whyte, DA Orlando, et al. Master Transcription Factors and Mediator Establish Super-Enhancers at Key Cell Identity Genes. *Cell*, 153:307–319, 2013.
- J Yu, MA Vodyanik, et al. Induced pluripotent stem cell lines derived from human somatic cells. *Science*, 318(5858):1917–1920, 2007.
- Y Zhang, T Liu, et al. Model-based Analysis of ChIP-Seq (MACS). *Genome Biology*, 9(9):R137, 2008.

6 APPENDIX I: SUPPLEMENTARY FIGURES

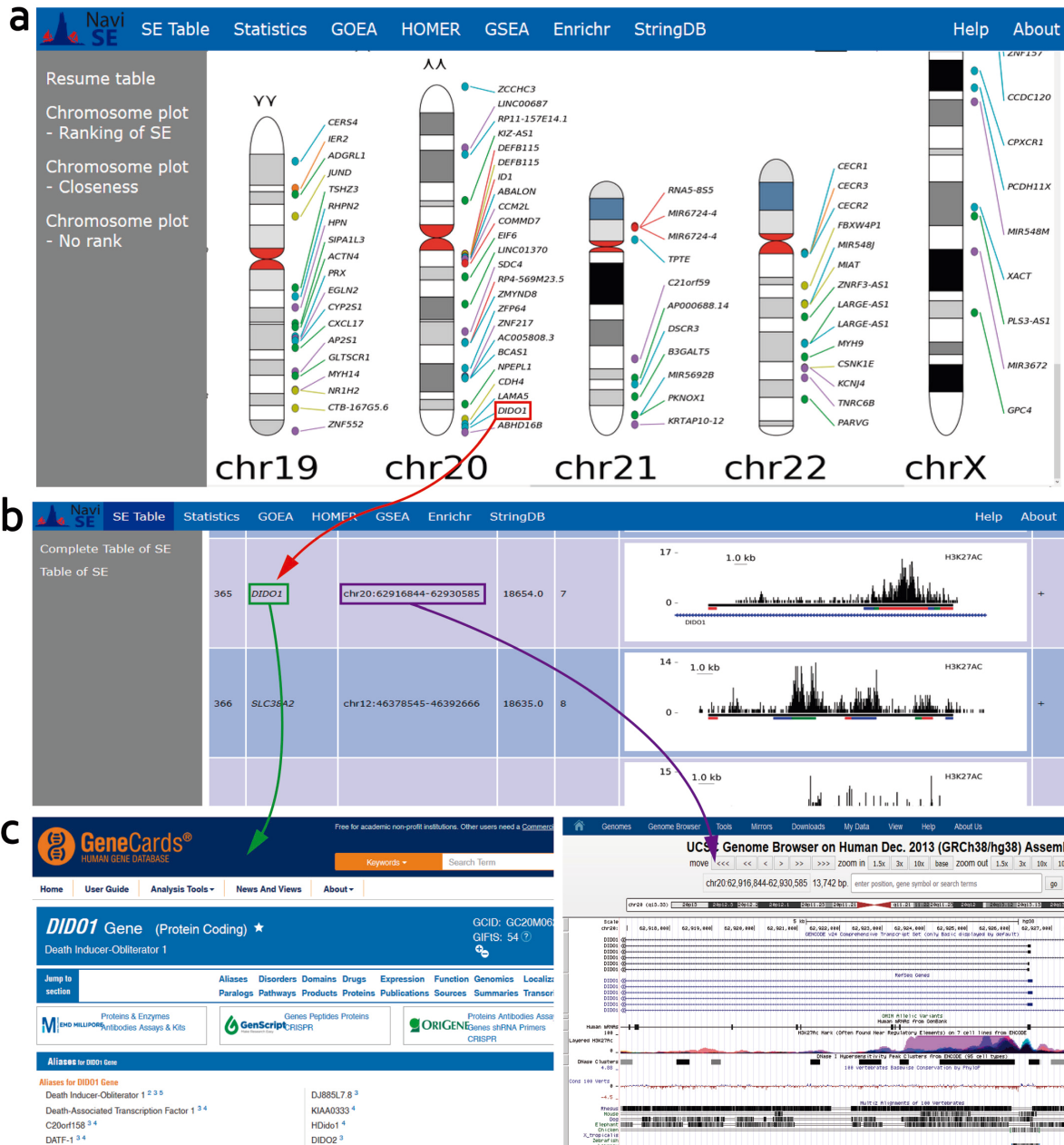


Figure S1. NaviSE GUI. All NaviSE windows contain a navigation bar on the top with links to all the results windows. On the left side there is a side *menu* bar with links to subsections of the active window. (a) The *main window* of NaviSE depicting the chromosomal plot in which the positions of all predicted SEs are mapped into a karyotype. Each SE in this window is a *hot-spot* with a link to the SE table. (b) Amongst other features, *SE table* contains the ranking of SEs, the names in the SE table linked to GeneCards (c), the chromosomal locations linked to UCSC Genome Browser (d), the SE score, the number of subpeaks, and, in the last column, the SE signal profile drawn with our GVT module.

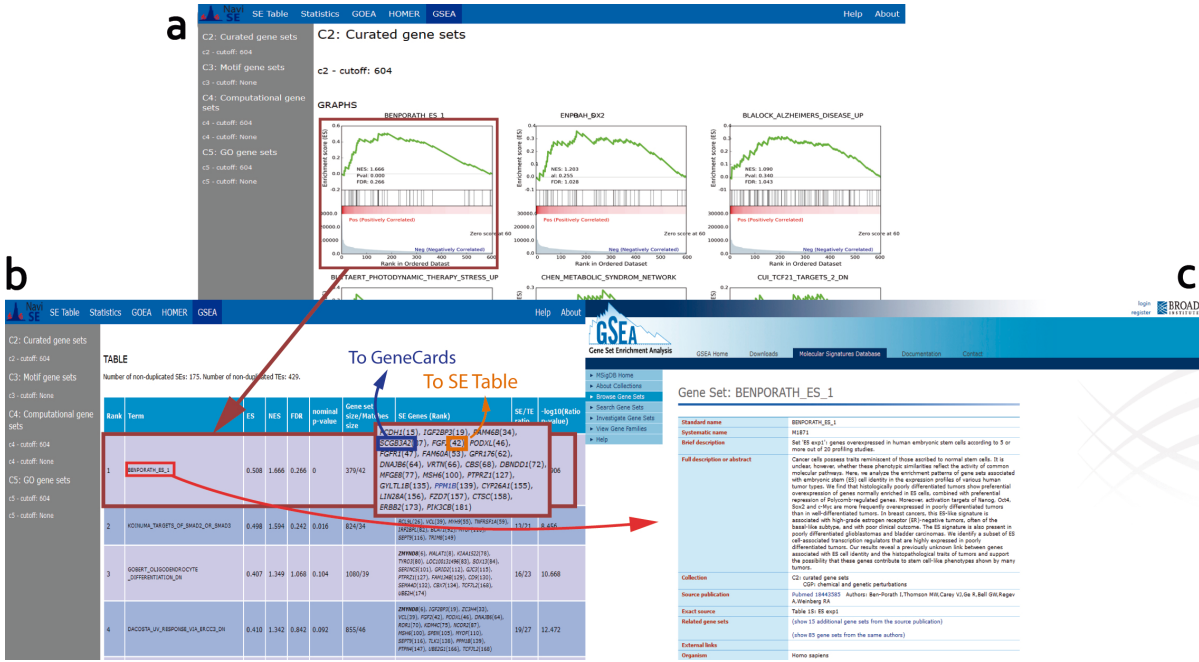


Figure S2. GSEA NaviSE GUI. (a) Window with interactive links to the corresponding GSEA terms in the GSEA table. (b) Table with the ranking of GSEA terms; each GSEA term, linked to GSEA website (c); statistical values such as Enrichment Score (ES), Normalized Enrichment Score (NES), FDR and p-values provided by GSEA; list of associated genes, linked to GeneCards with SE ranking value (in parentheses) linked to SE table window, described in Figure S1b.

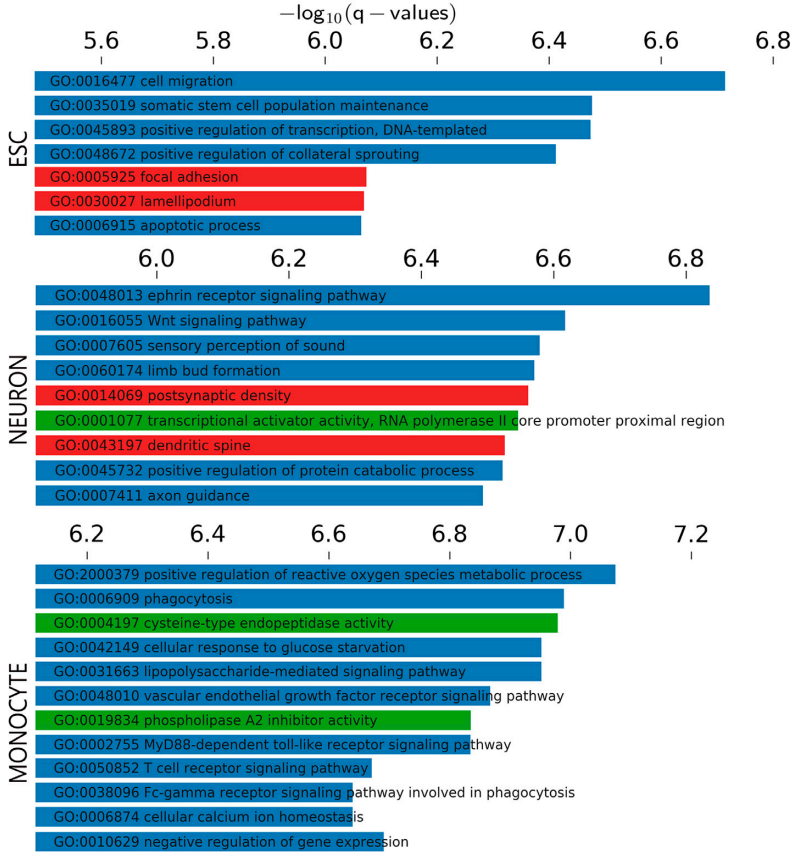


Figure S3. NaviSE GUI GOEA significant terms. Bar plots for each cell type depicting the most relevant and statistically significant terms for GOEA of the genes associated with SE predicted for H3K27ac. Red - cellular component, blue - biological process, green - molecular function.

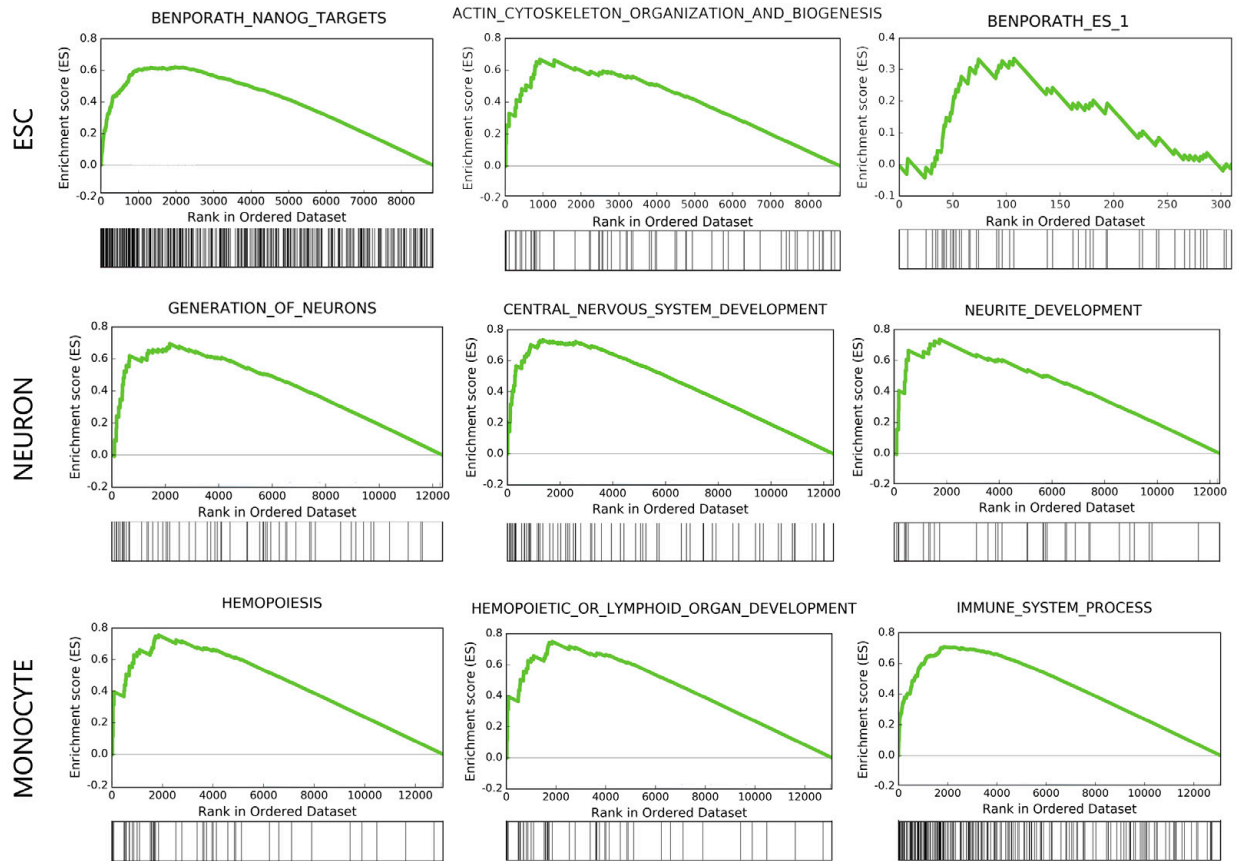


Figure S4. NavISE GUI GSEA most significant terms. GSEA profiles depicting three significant GSEA sets, from MSigDB, for each cell type for genes associated with SE predicted for H3K27ac. Each graph contains the typical GSEA profile alongside its positive matches in the bar below.

7 APPENDIX II: ORIGINAL ARTICLE

SOFTWARE

Open Access



NaviSE: superenhancer navigator integrating epigenomics signal algebra

Alex M. Ascensión^{1,2,3}, Mikel Arrospide-Elgarresta¹, Ander Izeta^{2*} and Marcos J. Araúzo-Bravo^{1,4*}

Abstract

Background: Superenhancers are crucial structural genomic elements determining cell fate, and they are also involved in the determination of several diseases, such as cancer or neurodegeneration. Although there are pipelines which use independent pieces of software to predict the presence of superenhancers from genome-wide chromatin marks or DNA-interaction protein binding sites, there is not yet an integrated software tool that processes automatically algebra combinations of raw data sequencing into a comprehensive final annotated report of predicted superenhancers.

Results: We have developed NaviSE, a user-friendly streamlined tool which performs a fully-automated parallel processing of genome-wide epigenomics data from sequencing files into a final report, built with a comprehensive set of annotated files that are navigated through a graphic user interface dynamically generated by NaviSE. NaviSE also implements an 'epigenomics signal algebra' that allows the combination of multiple activation and repression epigenomics signals. NaviSE provides an interactive chromosomal landscaping of the locations of superenhancers, which can be navigated to obtain annotated information about superenhancer signal profile, associated genes, gene ontology enrichment analysis, motifs of transcription factor binding sites enriched in superenhancers, graphs of the metrics evaluating the superenhancers quality, protein-protein interaction networks and enriched metabolic pathways among other features. We have parallelised the most time-consuming tasks achieving a reduction up to 30% for a 15 CPUs machine. We have optimized the default parameters of NaviSE to facilitate its use. NaviSE allows different entry levels of data processing, from sra-fastq files to bed files; and unifies the processing of multiple replicates. NaviSE outperforms the more time-consuming processes required in a non-integrated pipeline. Alongside its high performance, NaviSE is able to provide biological insights, predicting cell type specific markers, such as *SOX2* and *ZIC3* in embryonic stem cells, *CDK5R1* and *REST* in neurons and *CD86* and *TLR2* in monocytes.

Conclusions: NaviSE is a user-friendly streamlined solution for superenhancer analysis, annotation and navigation, requiring only basic computer and next generation sequencing knowledge. NaviSE binaries and documentation are available at: <https://sourceforge.net/projects/navise-superenhancer/>.

Keywords: Superenhancers, Next-generation sequencing, Parallel computing, Epigenomics, Computational biology, Graphics user interface, Signal algebra

*Correspondence: ander.izeta@biodonostia.org; mararabra@yahoo.co.uk

¹Computational Biology and Systems Biomedicine, Biodonostia Health Research Institute, 20014 San Sebastián, Spain

²Tissue Engineering Laboratory, Bioengineering Area, Biodonostia Health Research Institute, 20014 San Sebastián, Spain

⁴IKERBASQUE, Basque Foundation for Science, 48013 Bilbao, Spain

Full list of author information is available at the end of the article

Background

Superenhancers (SEs) are a novel class of transcription regulatory DNA regions with unusually strong enrichment for binding of transcriptional coactivators such as Mediator of RNA polymerase II transcription subunit 1 (MED1), activation histone marks such as H3K27ac, or cell and tissue-specific transcription factors (TFs) [1]. As a result, SEs represent large clusters of transcriptional enhancers that drive the expression of ‘master control’ genes that define cell identity. SEs differ from typical enhancers (TEs) for enclosing higher TF binding density and number of TF binding sites (TFBSs), which correlate with a much higher expression of their target genes [2]. Since SEs determine cell fate and gene expression regulation [3], they are related to altered expression of genes contributing to diseases such as Alzheimer or systemic lupus erythematosus [4]. Aberrant DNA methylation patterns in SEs, as well as SE-associated gene sets, have also been found to be altered in cancer [5–7].

Although protocols for computational prediction of SEs already exist [4], there is yet no tool that integrates all the processing stages from the raw data reads generated by the sequencer, through quality control and reads alignment, to peak estimation and peak stitching, ending with a fully annotated and interactive documentation of the results.

Furthermore, although SEs were initially predicted with MED1 [4] and activation histone marks such as H3K27ac, which has been proposed as a proxy for their estimation [2], the combination of several activation and repression epigenomics marks could help sharpen SE predictions. Therefore, we have designed NaviSE to use data with a wide range of chromatin status information, being able to process raw data from Assay for Transposase Accessible Chromatin (ATAC-seq) and DNase I hypersensitive sites (DHSs) experiments, apart from the usual CHIP-seq signals. In the case of other signals such as DNA methylation, NaviSE is prepared to integrate their information to perform SE predictions with the only condition that the user provides such data in bed or bam files, such as the bam files produced by the Parallel Processing Pipeline software for automatic analysis of Bisulfite Sequencing data (P3BSseq) [8].

On the other hand, there are no computational tools neither integrating several epigenomics signals simultaneously, nor performing signal algebra. Moreover, CPU resources and running-time are crucial for the high quantity of data produced by Next Generation Sequencing (NGS) technologies, hence another of the main demands in NGS software development is the parallelisation of the most time-consuming processes.

To meet all these demands, we have developed NaviSE, a user-friendly tool which automatically processes and integrates multiple genome-wide NGS epigenomics signals from various input file formats into an interactive

HTML report, built with annotations about SEs, such as associated genes, gene ontology (GO), graphs with metrics and statistical analysis, integrating all the data into the Graphical User Interface (GUI) to navigate through all the results. NaviSE parallelises the most relevant and time-consuming processes to optimise them, running multiple analysis in a significantly reduced amount of time. Finally, NaviSE is developed for users with working knowledge in informatics.

Implementation

Preprocessing of NGS files

Before the determination of SEs, NaviSE prepares the raw data, allowing multiple replicates and controls at once. The main steps for such preprocessing are as follows:

1. *Input format file recognition and file processing*: NaviSE recognizes multiple file formats, e.g., .sra, .fastq, .sam, .bam and .bed, and transforms an *upstream* format (.sra, .fastq, .sam) into a .bam file. In the absence of *upstream* files, *downstream* .bed files are also processed to .bam files.
2. *Alignments*: Performed by default with bowtie2 [9], .sam files are processed to .bam files by samtools. NaviSE also allows read alignment with MOSAIK [10], STAR [11] and BWA [12] aligners. Furthermore, users may generate their own .sam or .bam files with other aligners, and NaviSE will recognize these files for further processing.
3. *Quality control with FastQC*: NaviSE performs the quality analysis of the reads from the .fastq files using FastQC to create a report including several quality parameters, such as per base quality, GC content, *k*-mers distribution or presence of adapters.
4. *Combination of replicates and peak calling with MACS*: If there is more than one replicate or control, NaviSE will combine all the associated .bam files into one, and calculate the signal peaks with MACS (Model-based Analysis for ChIP-Seq) [13]. If control files are introduced for background correction, NaviSE configures MACS to use the control signal to calculate the peaks from the sample. Conversely, if no control is introduced, NaviSE configures MACS to use a pre-calculated background.

SE prediction and annotation

Once the data is preprocessed, a SE prediction and ranking is performed. SEs then are further analysed in search of SE related genes, DNA sequence motifs, GO terms or statistical estimations.

1. *Epigenomics signal algebra*: In case more than one epigenomic signal was used to predict SEs, NaviSE integrates all the signals to improve the SE prediction.

The way in which different epigenomic signals are combined is defined by the names of the signal data files $Sig \in \{H3K27ac, H3K4me1, H3K4me3, H3K9me3, H3K27me3, ATAC-seq, DHS, \dots\}$ separated by signal operators $Ope \in \{AND, OR, NOT, XOR, +, - SYM\}$.

The way these algebra operators have been adapted to operate over pairs of genomic signals is illustrated in Fig. 1. To invoke this algebra, NaviSE is called writing these signal and operators as additional arguments in the command line:

$$Sig_1 Ope_{1,2} Sig_2 Ope_{2,3} Sig_3 Ope_{3,4} Sig_4$$

where Sig_i is the name of the file containing the epigenomics data of a type of signal i , and $Ope_{i,i+1}$ is the pairwise signal operator applied to combine i and $i+1$ signals.

When performing ‘epigenomics signal algebra’, NaviSE picks the first pair of signals separated by each operator starting from the left ($Sig_1 Ope_{1,2} Sig_2$). Once an operation is processed, its results are combined with the next signal using the next operator ($(Sig_1 Ope_{1,2} Sig_2) Ope_{2,3} Sig_3$). This process continues from left to right side recursively until the last signal identifier is reached. To speed up the process, for each pair

of signals NaviSE searches first all their overlapping regions and performs the signal operator only over these regions.

2. *SE prediction*: To predict the SEs in a sample, NaviSE performs the *stitching* of MACS peaks which fall within a threshold distance, using our own implementation of the algorithm developed by Young’s lab [2], in which MACS peaks (inferred as enhancers) are stitched according to a constant distance (12.5 kb by default) criterion algorithm, in case the distance between the end of one MACS peak and the start of the following peak is less than the established threshold, they are *stitched* as a single peak. Then, NaviSE ranks the *stitched* peaks with a score based on the measured signal level within the *stitched* region.

NaviSE assigns a score to each *stitched enhancer*, considering that a *stitched enhancer* with a higher number of bam reads has a higher SE predictive value. Thus, to build the SE ranking, NaviSE takes the raw reads from the .bam files, and for each *stitched enhancer* it collects all the reads over the *stitched enhancer* support. This support is defined by the DNA sequence lying between the *stitched enhancer* start, $STIT_{start}$, and the *stitched enhancer* end, $STIT_{end}$, nucleotide positions. Then, we define the *stitched enhancer* count, $Count_{STIT}$, as the cumulative sum of the bam reads throughout the *stitched enhancer* support:

$$Count_{STIT} = \sum_{i=1}^{N_{reads}} \sum_{j=STIT_{start}}^{STIT_{end}} read_i(j) \tag{1}$$

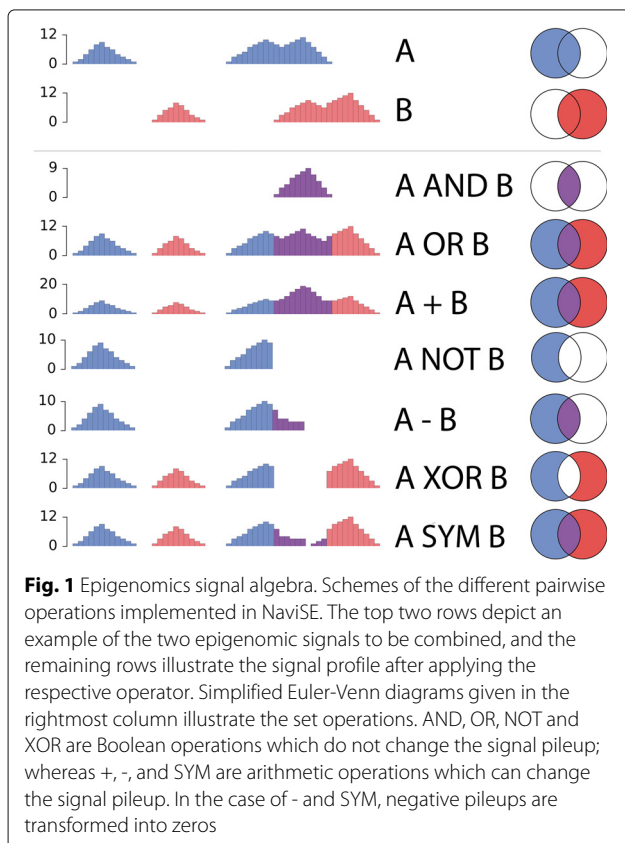
where $read_i(j)$ indicates whether a bam read i , from the set of N_{reads} , lies at the position j of the *stitched enhancer* within the support $[STIT_{start}, STIT_{end}]$. Therefore, $read_i(j) = 1$ if a nucleotide of the bam read i is mapped to the location j of the *stitched enhancer*, and $read_i(j) = 0$ otherwise.

Then, the SE ranking, r , is defined as the sorted list of $Count_{STIT}$ in descending order:

$$r = \text{sort}_{\downarrow} \{Count_{STIT}\} \tag{2}$$

thus, Eq. 2 assigns position one in the ranking to the *stitched enhancer* with the highest $Count_{STIT}$, position two in the ranking to the *stitched enhancer* with second highest $Count_{STIT}$, etc. until we reach the *stitched enhancer* with the lowest $Count_{STIT}$.

The next step is the determination of the SE threshold (θ_{SE}), the position of the ranking for which the *stitched enhancers* whose rank is below θ_{SE} will be considered as SEs, and TEs otherwise. To determine θ_{SE} , we scale both $Count_{STIT}$ and r between 0 and 1. Then, we determine θ_{SE} as the position of r whose slope is nearest to 45°.



3. *SE gene assignment*: Once the SE locations are determined, each SE is assigned a gene by proximity with the closest transcription start site (TSS). NaviSE also includes information about genes overlapping the SE or genes proximal to each SE.
4. *Subpeak annotation*: The SEs and TE subpeaks have been shown to act synergistically within the SE despite being individual and independent structures [14]. To provide detailed information about the SE subpeaks structure and location, NaviSE performs a structural annotation of the subpeaks that represent each SE. The annotation contains the following parameters:
 - Number of subpeaks, *loci* and TSS locations.
 - Association to TSSs: Due to the TSS specific regulation role, a SE inside a TSS might not exert the role of SE itself. Thus, to understand the regulatory role of the SEs, it is important to resolve their association to TSSs. This analysis is portrayed by two related values: (i) the *Percentage OUTS*, which is the percentage of subpeaks outside the range of the user-defined distance within the TSS, and (ii) the *Enhancer Type*, a classification of the SE according to *Percentage OUTS*. The categories assigned to *Enhancer Type* are labeled as *Pure* if all the subpeaks are outside the TSS, *Only TSS* if all the subpeaks lay within the TSS, and *Mixed* if there are both types of subpeaks.

5. *Automatic generation of SE peak distribution profiles*: To visualize the SE peak distribution we have implemented in NaviSE our own Genome Viewer Tool (GVT). With this tool, two snapshots at *near* and *far* distances for each SE are portrayed, which are shown in the *SE table* window of the final report. NaviSE dynamically calculates the optimal range for each snapshot, based on the width of the SE. With *near* shot the user is able to determine the morphology of the SE, and with *far* the user is able to locate the SE in its genomics surroundings. In each snapshot both the location of the SE and the enhancer peaks determined by MACS are shown.
6. *HOMER motif finding*: SEs enclose high number of TFBSs [2]. Therefore, identifying such TFBSs is important for SE annotation. To find motifs of regulatory elements (mainly TFs) that are specifically enriched in the *loci* of SEs, relative to the *loci* of TEs, NaviSE uses the Hypergeometric Optimization of Motif Enrichment (HOMER). As a result, NaviSE generates in the final report a *HOMER table*, which includes motifs enriched in SEs, and a list of *de novo*

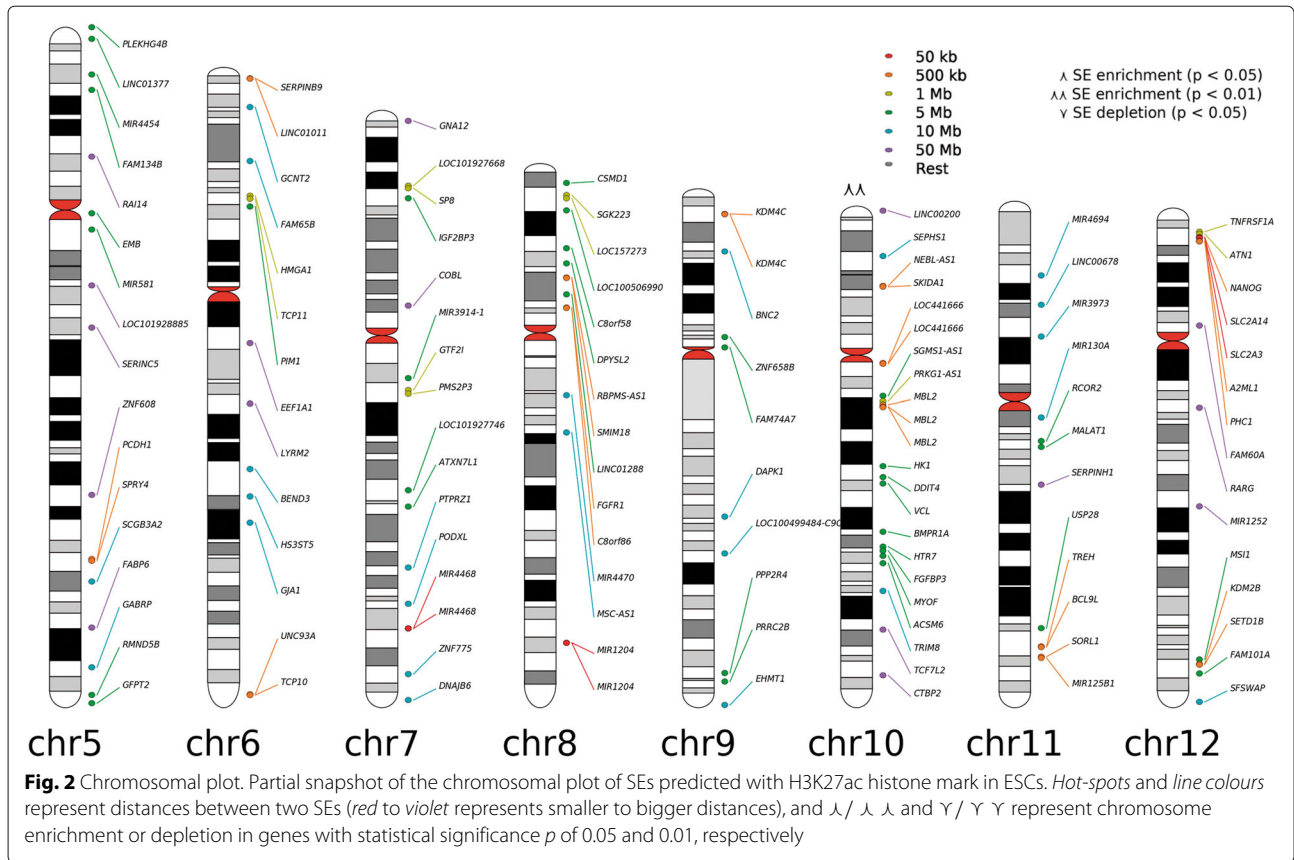
motifs for which their respective binding elements are predicted by HOMER.

7. *Gene Ontology Enrichment Analysis (GOEA)*: To predict the functionality of the SEs, based on the closest gene of each SE determined by HOMER, NaviSE uses *goatools* [15].
8. *Pathways and protein-protein interaction annotation*: To obtain annotation of TFs and pathways related to SEs, NaviSE uses *Enrichr* [16]. To obtain protein-protein interaction (PPI) networks of SEs, NaviSE uses the database of PPIs *String* [17]. Results from *Enrichr* and *String* are processed and integrated into the final report to be navigated through NaviSE GUI for an easier interpretation for the user.
9. *NaviSE GUI*: To navigate throughout all the results, we have implemented an interactive chromosomal plot (Fig. 2) that represents the SE location in a karyotype; alongside with graphs that depict statistical values and properties related to SEs (shown in “Results” section), as well as information related to GOEA or *Enrichr*.

Chromosomal plots are designed to include *hot-spots* with links to the elements of the *SE table* from the final report, which are activated when the user navigates with the mouse over the gene names on the chromosomal plot. To enhance the usability of this feature, NaviSE generates three types of chromosomal plots:

- Enrichment plot: it shows the *loci* location and the chromosome enrichment or depletion.
- Rank plot: it shows *loci* coloured according to their rank in the *SE Table*. Several percentiles are represented based on the rank of the SE, and SEs falling within a percentile will be coloured correspondingly.
- Closeness plot: it represents the proximity between SEs, according to which SEs will be coloured. This plot is highly useful to discern clusters of SEs that look overlapped. For the ordered list $\{SE_1, SE_2, \dots, SE_{a-1}, SE_a, SE_{a+1}, \dots, SE_{c-1}, SE_c\}$, of c SEs within a chromosome, for which each SE_k support is defined by its start ($SE_{k,start}$) and end ($SE_{k,end}$) *loci* positions, the closeness of a SE_k is its distance $C(SE_k)$ to the closest SE, determined by the following expression:

$$C(SE_k) = \begin{cases} SE_{2,start} - SE_{1,end} & \text{if } k = 1 \\ SE_{c,start} - SE_{c-1,end} & \text{if } k = c \\ \min(SE_{a+1,start} - SE_{a,end}, \\ E_{a,start} - SE_{a-1,end}) & \text{otherwise} \end{cases} \quad (3)$$



In all these chromosomal plots a probability p determining whether a chromosome is enriched (marked with \wedge for $p < 0.05$ and $\wedge\wedge$ for $p < 0.01$) or depleted (marked with Υ for $p < 0.05$ and $\Upsilon\Upsilon$ for $p < 0.01$) with SEs is calculated by a binomial approximation of the hypergeometric distribution ($h(k; K, n, N) \rightarrow b(k; K, z) ; z = \frac{n}{N}$) [18], where N is the number of genes in the whole genome, K is the number of SEs in all chromosomes, n is the number of genes in that chromosome, and k is the number of SEs in that chromosome.

10. *Gene Set Enrichment Analysis (GSEA)*: To obtain additional functional annotation of SEs, NaviSE performs the GSEA [19] from SE-associated genes, using gene sets from the Molecular Signatures Database (MSigDB).

Statistics of the comparison between TE and SE

Although both SEs and TEs derive from MACS peaks, they structurally differ for having higher peak density. To illustrate the differences between SEs and TEs, NaviSE shows in the final report a collection of metrics and plots depicting the differences between them. Among the most important plots are:

1. **Ranking of SEs by the order of SE score**: It is the plot of $Count_{STIT}$, given by Eq. 1 vs r , given by Eq. 2. It typically shows a *hockey stick* shape, with the inflection point marking the boundary between SEs and TEs, θ_{SE} .
2. **INs and OUTs**: It shows statistics about the percentages of SEs and TEs that lay within a TSS or not. This might be interesting if a sample contains an elevated percentage of SEs within TSS, as some of these SEs might be misinterpreted as promoter signals.
3. **SE vs TE length distribution**: It shows the distribution of SE and TE length and pileup in a double histogram and a scatter plot. The histogram lying on the X-axis of the scatter corresponds to the length of SEs and TEs; and the histogram on the Y-axis corresponds to the pileup. This graph is complementary to the ranking of SEs by SE score, to shed light on the population of SEs and TEs.
4. **SE vs TE subpeak length distribution**: This graph contains the same elements than the SE vs TE length distribution graph, although showing the distribution of enhancers inferred by MACS.
5. **Number of subpeaks (for number of bins $N = 10$ and $N = 20$)**: It shows the distribution of subpeaks each SE or TE has.

Generation of the NaviSE report

The final step of NaviSE is the generation of an HTML report, in which all the results from the analysis are gathered and presented within several windows, each of which contains interactive links both to external website which provide the user with further information about the SEs, as well as to other internal HTML pages created by NaviSE within the report. The content of this report is discussed in detail in the “Results” section.

Parallelisation implementation

The algorithm of parallelisation developed in NaviSE constitutes a significant improvement of performance in the analysis of NGS samples compared to non-parallelised pipelines. NaviSE determines the optimal number of processes, k , compatible with the computer resources as Luu et al. do in [8]. Such resources are the parallel processing capability of the computer measured as the number of cores, C , and the total main memory, M , in Gigabytes (GBs). NaviSE optimizes automatically, for each processing task i , the number of threads, k_i , in which the task i will be parallelised by the expression:

$$k_i = \min(C, C_u, \lfloor M/m_i \rfloor, l_i) \quad (4)$$

where C_u is the maximum number of cores reserved by the user to run NaviSE, m_i is the memory, measured in GBs, needed to run one process in task i , $\lfloor \cdot \rfloor$ is the floor operator and l_i is the cardinal of $D_i = \{d_1, d_2, \dots, d_m\}$ which is the set of *chunks* of distributed data elements to be processed in task i . If $l_i > k_i$, the first k_i chunks are distributed to k_i threads. The distribution of information (number of chromosomes for stitching, SE peak distribution profiles for GVT, number of gene sets for GSEA) to be parallelised is based on a cyclic algorithm, implemented in Python, with the following outline: For the ordered set $S_i = \{s_1, s_2, \dots, s_n\}$ of information elements, the set $P_i = \{1, \dots, k_i\}$ of processes, and for the set D_i of data (chromosomes, positions on a list, gene sets) to be distributed across processors, we define D_{pi} as the *chunk* of data of the task i that is assigned to each processor p :

$$D_{pi} = \{d_j \mid \forall d \in D_i, p \in P_i, j \in \{1, \dots, l_i\}, j \bmod k_i = p\} \quad (5)$$

where mod is the module operator. Once the *chunk* D_{pi} is constructed, the subset of information elements $S_{D_{pi}} \subset S_i$ will be defined depending on the task i which is being parallelised. The list of parallelised tasks in NaviSE is $i = \{\text{STIT}, \text{GVT}, \text{GSEA}, \text{HOMER}\}$. In the case of parallelisation of SE prediction by stitching (STIT), the set of peak coordinates from MACS (S_{STIT}) is divided into k_{STIT} files, calculated with Eq. 4, with $m_{\text{STIT}} = 3$ GBs. Here, $D_{\text{STIT}} = \{1, 2, 3, \dots, X, Y\}$ chromosomes, $D_{p,\text{STIT}}$ represents the sets of chromosomes that will be processed

in each $p \in P$ calculated by Eq. 5, $S_{D_{p,\text{STIT}}}$ is the *chunk* of $s \in S_{\text{STIT}}$ peaks which share the same chromosome from each set of chromosomes from $D_{p,\text{STIT}}$. For a better understanding of the process, an example is developed in Fig. 3.

In the case of SE signal profile generation with GVT, $S_{\text{GVT}} \equiv D_{\text{GVT}}$, is the set of SE *loci*. Hence $D_{p,\text{GVT}}$ contains all the *loci* that fulfill Eq. 5, based on k_{GVT} calculated with Eq. 4 with $m_{\text{GVT}} = 2$ GBs.

In the case of GSEA parallelisation, D_{GSEA} is the set of combinations (GSEA signatures \times GSEA cutoffs) and S_{GSEA} is the set genes associated to SEs and TEs up to the corresponding GSEA cutoff. Therefore, $D_{p,\text{GSEA}}$ contains all the combinations that fulfil the Eq. 5, based on k_{GSEA} calculated with Eq. 4 with $m_{\text{GSEA}} = 2$ GBs.

The parallelisation of all these tasks has been implemented with the *multiprocessing* module of Python. In the case of HOMER parallelisation, we took advantage of the capabilities already implemented in HOMER, with the number of processes k_{HOMER} , optimized by Eq. 4, with $m_{\text{HOMER}} = 2$ GBs.

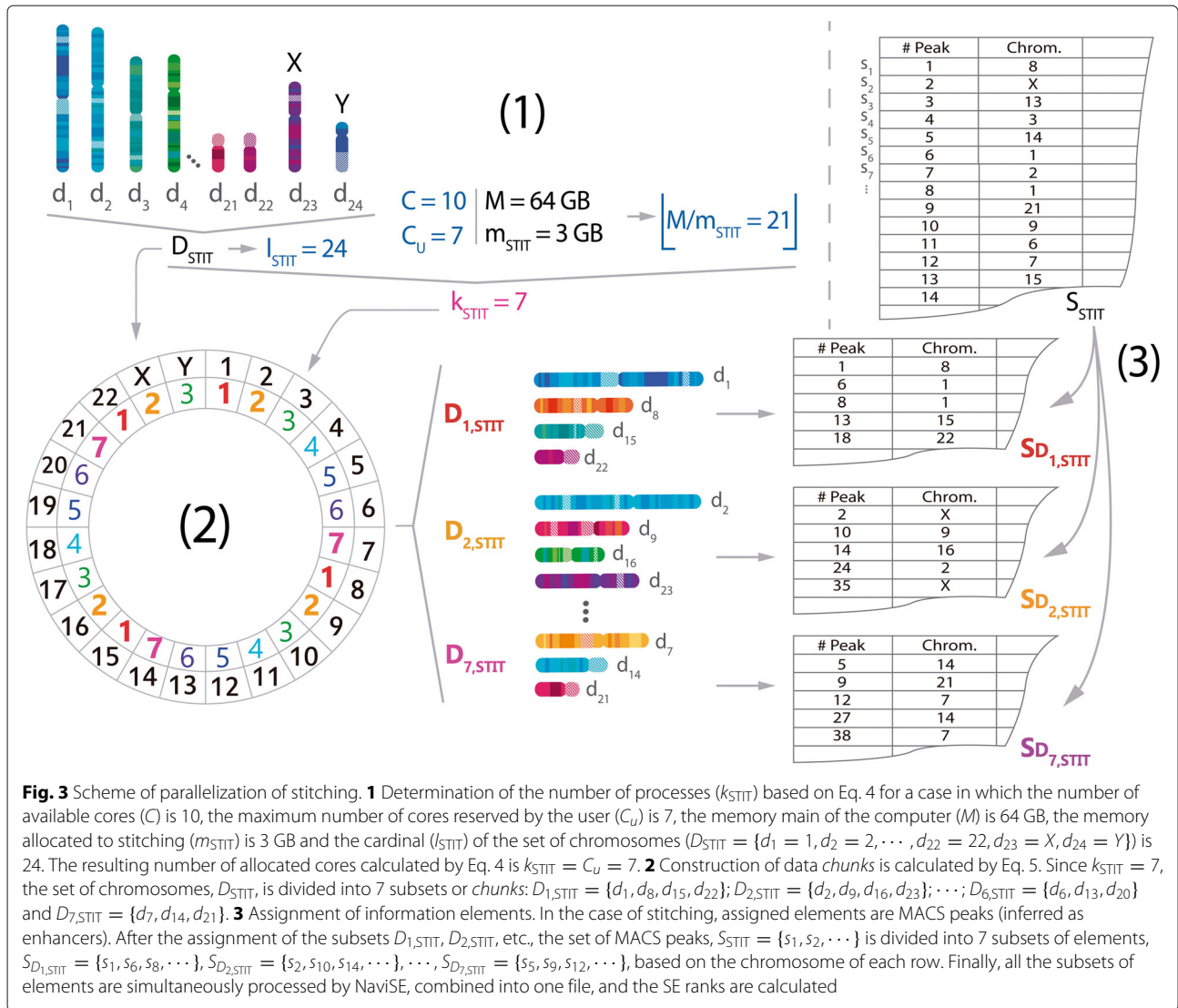
Results

To illustrate the performance of NaviSE, we have selected H3K27ac histone mark whose raw signal data has been downloaded from the GEO database [20] for three cell types: human Embryonic Stem Cells (ESC) (GSM663427, with control GSM605335), monocytes (MON) (GSM1003559 with control GSM1003475) and neurons (NEU) (GSM2072642, with control GSM2072639). For other analysis, we also used H3K4me1 (GSM409307) and H3K4me3 (GSM409308) from ESCs.

HTML report generation

The output of NaviSE for each experiment is a collection of HTML linked pages whose main page contains dynamic graphical elements, namely, a blue horizontal ribbon with links to all the HTML pages from the report, detailed below; a grey sidebar by which the user can access the different subsections; and a window in which the results are displayed.

The *main window* contains basic information about the analysis and different chromosomal plots, defined in the point 9 of “SE prediction and annotation” section, represented in the chromosomal plot snapshot of Fig. 2. The chromosomal plot includes links to the SEs in *SE Table* window of the final report, which includes general information about each SE (genomic location, number of subpeaks, SE score), alongside with a snapshot of the SE genomic signal profile, included for visual evaluation of the SE quality, together with the quantitative SE score. The *SE Table* columns referring to gene names and genomic location include, respectively, a link to GeneCards site [21] and UCSC Genome Browser [22], as shown in Fig. 4.



Statistics window implements a series of graphs which allow the user to obtain information related to the SEs in the sample. Some of those graphs are analysed thoroughly in the corresponding Analysis of different cell lineages “Results” section.

GOEA window includes the results from the GOEA. At first, a barplot shows the significant terms from GO categories (biological process, cellular component and molecular function) which, upon clicking, will lead to a Directed Acyclic Graph (DAG) of the GO terms associated with the significant term, each of which contains the related genes associated to that term. Below the barplot, there is a table that leads to the DAG for the corresponding GO term, which includes values such as enrichment ratio of the predicted cell population, and the False Discovery Rate (FDR) for each term.

Similarly, the GSEA window (Fig. 5) contains several graphs depicting the GSEA profile of the significance of the analysis, for each signature (group of gene sets) and threshold. Clicking on a graph leads to its corresponding information element on a table below, which contains several related values, such as the significant GSEA term, related SE genes, and statistical values linked to the GSEA term such as Enrichment Score (ES), Normalized Enrichment Score (NES), FDR and p -values provided by GSEA, which are further described in Additional file 1.

HOMER window shows the results from the motif analysis by the HOMER tool, which includes two ranked tables, one for known motifs and another one for *de novo* motifs. ‘Known motifs’ table contains a LOGO image for each motif and the name of the TF or binding protein using such binding motif. It also includes the percentage of SE and TE sequences that has such motif, and a p -value

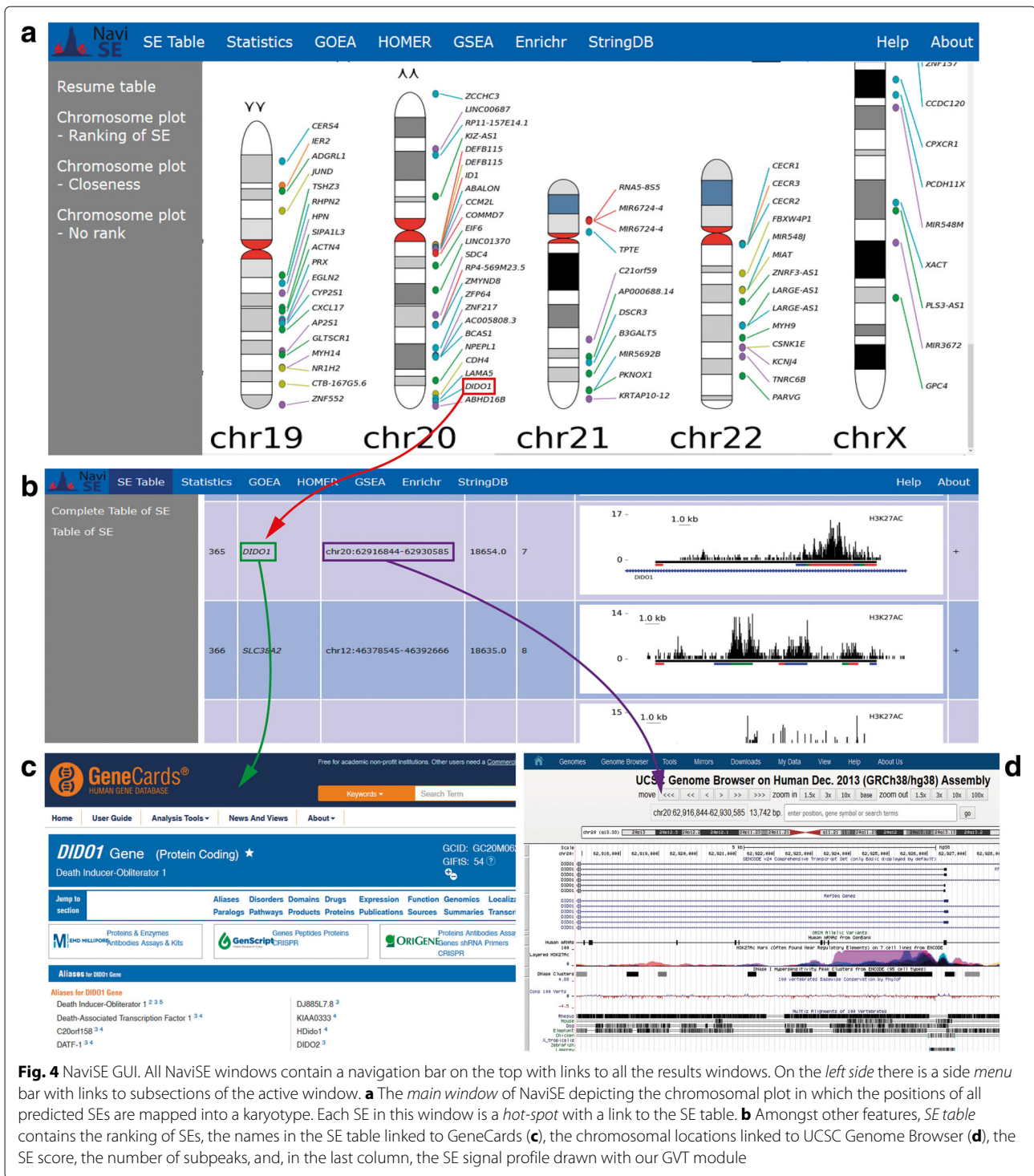


Fig. 4 NaviSE GUI. All NaviSE windows contain a navigation bar on the top with links to all the results windows. On the *left side* there is a *side menu* bar with links to subsections of the active window. **a** The *main window* of NaviSE depicting the chromosomal plot in which the positions of all predicted SEs are mapped into a karyotype. Each SE in this window is a *hot-spot* with a link to the SE table. **b** Amongst other features, *SE table* contains the ranking of SEs, the names in the SE table linked to GeneCards (**c**), the chromosomal locations linked to UCSC Genome Browser (**d**), the SE score, the number of subpeaks, and, in the last column, the SE signal profile drawn with our GVT module

that measures the statistical significance of the association of the SE with such motif. The ‘*de novo*’ table includes motifs predicted by HOMER to bind elements differentially in SEs and TEs. Upon clicking on each element in the ‘*de novo*’ table, NaviSE redirects to a HOMER-generated page that includes more information about the motif.

Finally, *StringDB* and *Enrichr* windows show, respectively, PPI networks from SEs at different confidence values; and results from Enrichr website including TFs related to SEs, cell or tissue specification or metabolic pathways linked to the SE population. Each subsection includes a barplot of the significant terms which link to

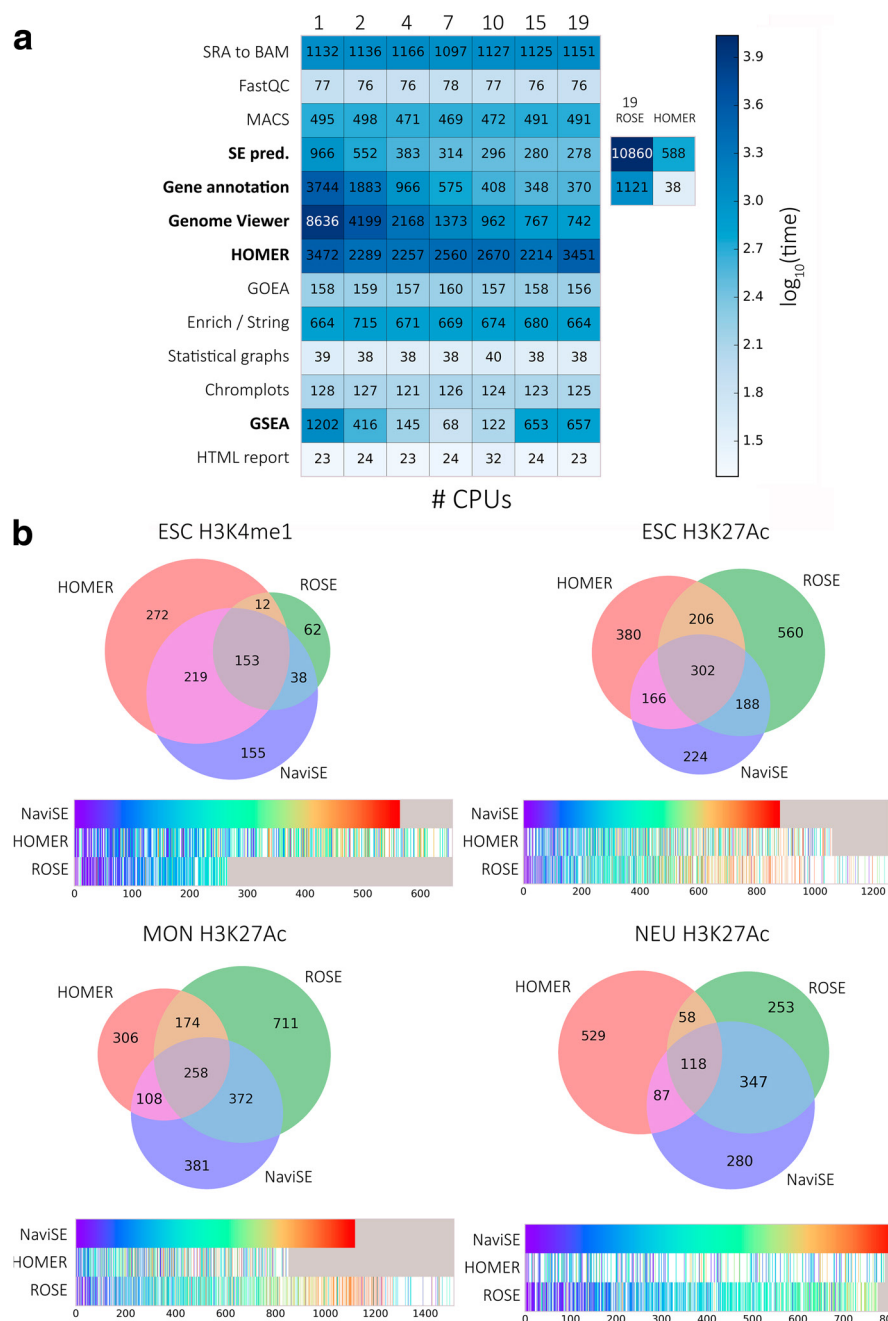


Fig. 6 NaviSE performance comparisons. **a** NaviSE CPU running time for different numbers of CPUs. Heatmap of the processing time for each NaviSE process for different numbers of CPUs, written on top. Tasks parallelised by NaviSE are highlighted in *bold* typeface. For the SE prediction and gene annotation, running times of ROSE and HOMER on 19 CPUs are also provided. **b** SE prediction similarities among different software. For each cell type and histone mark, the Euler-Venn diagram with the number of commonly predicted SEs is represented on top and the comparisons among the SE ranking generated by the different software at the bottom. The rank of each SE predicted by NaviSE is colour-coded (*blue* colours indicate higher positions in the rank and redder colours lower positions). Each NaviSE SE is mapped onto HOMER and ROSE SE ranking tracks in the position predicted by HOMER and ROSE for such SE, with the colour codification corresponding to the ranking predicted by NaviSE. SEs predicted by other software that are not predicted by NaviSE appear in white. *Grey* boxes mark the indexes for which a rank in a predictor has exhausted its number of predicted SEs in comparison to the maximum rank predicted by the three software {HOMER, ROSE, NaviSE}

predictions we have designed a graphical representation that allows us to track the ranking of each SE predicted by each software in comparison with the ranking predicted by NaviSE. This representation shows that the rank of the score of the SEs is very similar among all of the predictors (ranking bars in Fig. 6b). A detailed explanation of prediction divergences between different software, as well as between epigenomic combinations, is provided with an example with ESCs at “NaviSE epigenomics signal algebra is able to predict SEs with sharper signals” “Results” section.

SE prediction of different cell lineages

To assess the capabilities and performance of NaviSE, we have run several real datasets from different species (human and mouse), histone marks (H3K27ac, H3K4me3 and H3K4me1), and cell types (ESC, MON and NEU), using the hg38 human genome version.

Main page, SE table, and Statistics

Using the same default parameters with H3K27ac histone mark, the NaviSE analysis for the different cell lines yielded a wide range of SEs (n_{ESC} : 664, n_{NEU} : 1073, n_{MON} : 1235). The signals of the most important SEs are shown in the Fig. 7 and the main statistics for each cell type are depicted in the Fig. 8.

The distribution of subpeaks varies considerably between SEs and TEs. TE subpeak distribution follows a Zipfian-like distribution in all the analysed cell lines, that is, most of the samples contain only 1 subpeak, and the number of samples that contain higher amount of subpeaks goes down at a rate of $\sim 50\%$ of the previous number of subpeaks; whereas the SE distribution might follow a χ^2 distribution or a normal distribution. In the case of ESCs, the maximum of subpeaks is between 5 and 7, whereas in NEU and MON the distribution is uniform between 6 and

14 subpeaks, with a considerable amount of SEs having more than 20 subpeaks.

The differences in length distribution between TEs and SEs are apparent in all samples. Interestingly, TEs usually show a bi or trimodal distribution with maxima at ~ 100 , ~ 1000 or $\sim 10,000$ nt in all the analysed cell types, whereas SEs show a monomodal normal-like distribution with means around 25,000 - 50,000 nt. On the other hand, subpeak distribution shows no significant differences between SEs and TEs, both in length and pileup.

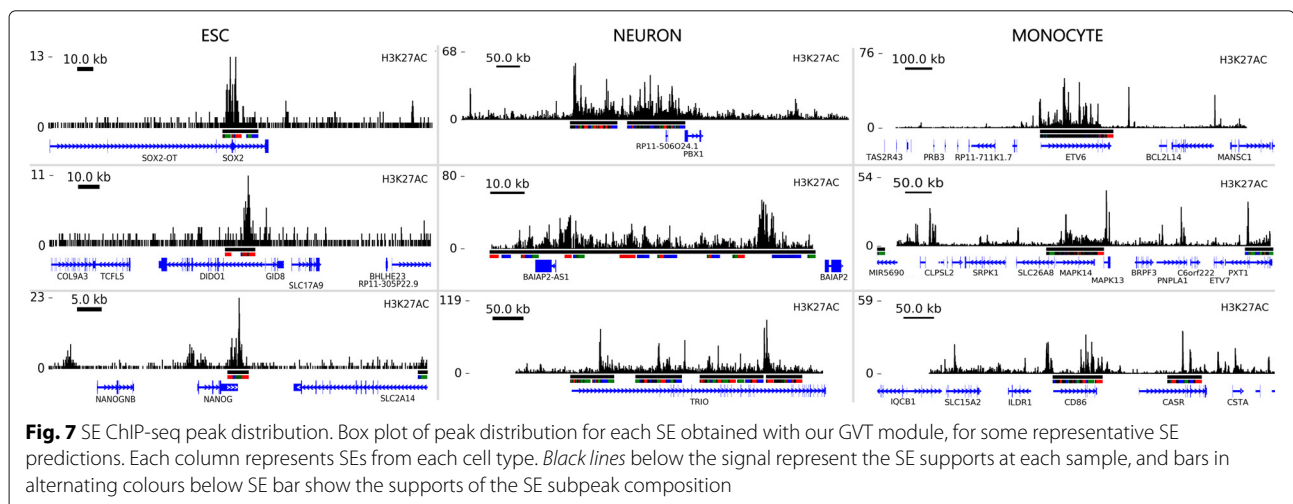
HOMER analysis

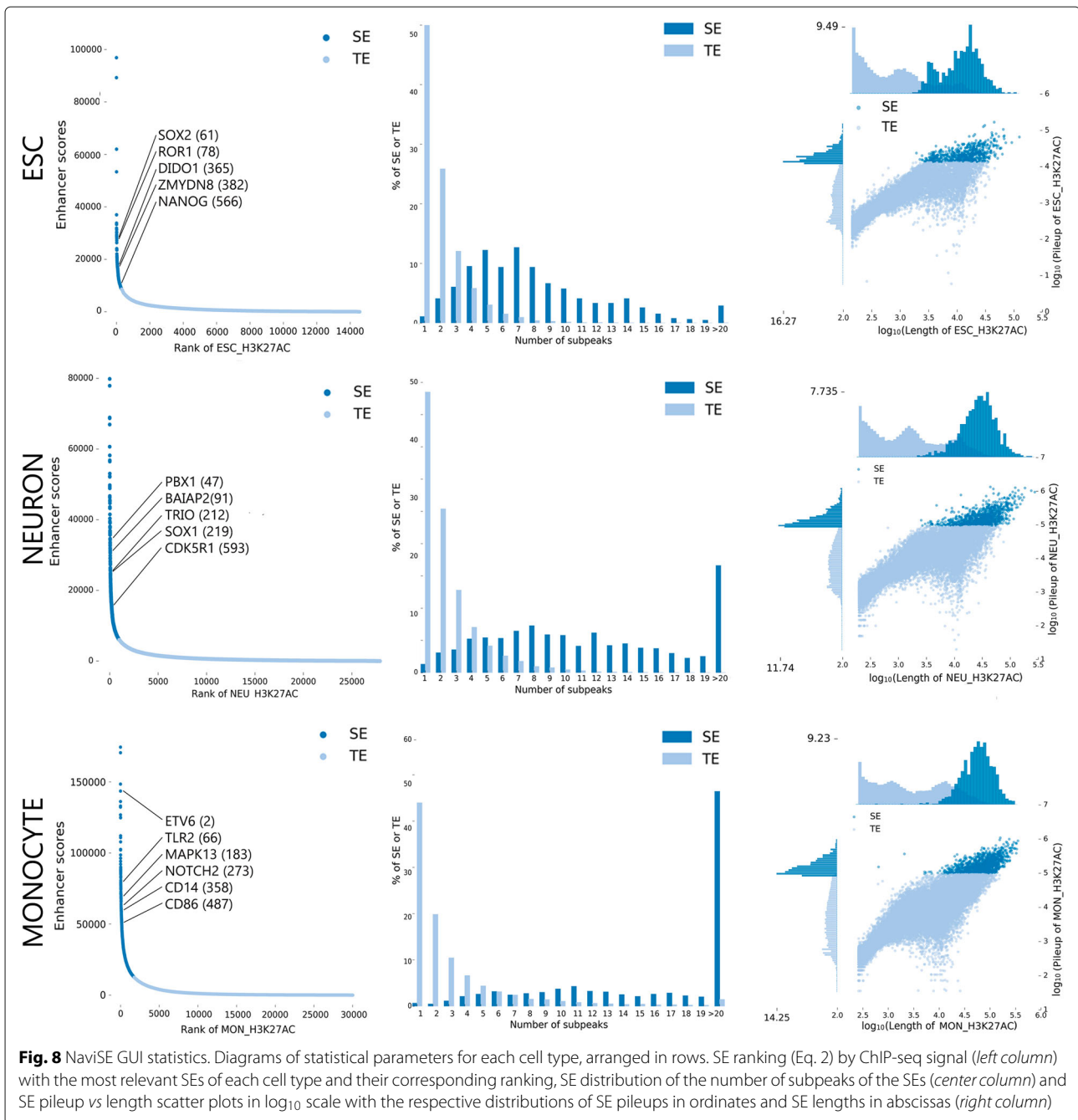
The results of the most relevant TFs revealed by HOMER are shown in Table 1. Although all three cell lines showed shared TFs such as *TCF3*, each cell type contained a set of cell-specific TFs. For instance, ESC contained *NKX2-2* and *NKX2-5*, involved in heart and nervous development; NEU contained *RXR*, involved in neural development, *NR5A2*, involved in embryonic development and, interestingly, *RUNX1*, thought to be involved in hematopoiesis. Finally, MON contained *GATA2* and *GATA1*, the first closely related to hematopoiesis and the second involved in the switch of fetal hemoglobin to adult hemoglobin.

GOEA and GSEA results

GOEA and GSEA are represented in Figs. 9 and 10 respectively. Both results are related, as the signatures used for GSEA contain sets of genes related to GO sets. Both analysis show correlation of functions to each cell type.

For ESC, the most relevant GO terms are related to protein expression (*positive regulation of transcription*), rearrangement of cellular morphology (*focal adhesion, lamellipodium*) or pluripotency (*somatic stem cell population maintenance*). As for GSEA, significant terms are related to master TFs of ESCs, such as *NANOG*,





or cytoskeletal reorganization. Among the predominant genes, most of them repeated in several functional terms, we remark *ROR1* (which modulates neurite growth and is highly expressed during early embryonic development [23]), *ZIC3/5* (involved in the formation of right/left axis during development, and direct activator of *NANOG* promoter in ESC [24]) or *SOX2* (one of the Yamanaka’s reprogramming TFs, used for the induction of pluripotency, as well as a core pluripotency factor in ESC [25]).

Regarding NEU, the most relevant GO terms are related to neural development (*ephrin signaling*, *Wnt signaling*

pathway, *dendritic spine*, *axon guidance*). As for GSEA, three relevant terms are *generation of neurons*, *neuron differentiation*, and *neurite development*. Three highly ranked genes in these GSEAs are *CDK5R1* (neuron-specific activator of cyclin-dependent kinase 5, required for proper development of the central nervous system, also found essential for oligodendrocyte maturation and myelination [26]), *BAIAP2* (brain-specific angiogenesis inhibitor binding protein, might be related to neural growth-cone guidance, dendritic spine development and NMDA receptor regulation [27]) and *PBX1* (regulates

Table 1 The most relevant TFs, and their binding motifs for all cell types obtained from HOMER analysis. *p*-values are presented in their integer logarithmic form ($pP\text{-val} \equiv -\log_{10} P\text{-val}$)

ESC			NEU			MON		
TF	Motif	pP-val	TF	Motif	pP-val	TF	Motif	pP-val
NKX3-2		121	TCF3		142	TCF3		142
NKX2-2		83	TBX21		132	TEAD2		105
NKX2-5		77	RXR		124	NPAS2		90
ESRRA		75	RUNX1		124	GATA2		85
TBX5		75	NR5A2		122	GATA1		83

differentiation and survival of certain neurons, and is impaired in Parkinson's disease [28, 29]).

Regarding MON, the most relevant GO terms are related to specific functions of monocytes involved in immune response (*phagocytosis, T cell receptor signaling*

pathway, MyD88-dependent toll-like receptor signaling pathway, lipopolysaccharide-mediated signaling pathway). As for GSEA, three relevant terms are *T cell receptor signaling pathway, reactome immune system and immune system process*. Genes shared by several GO

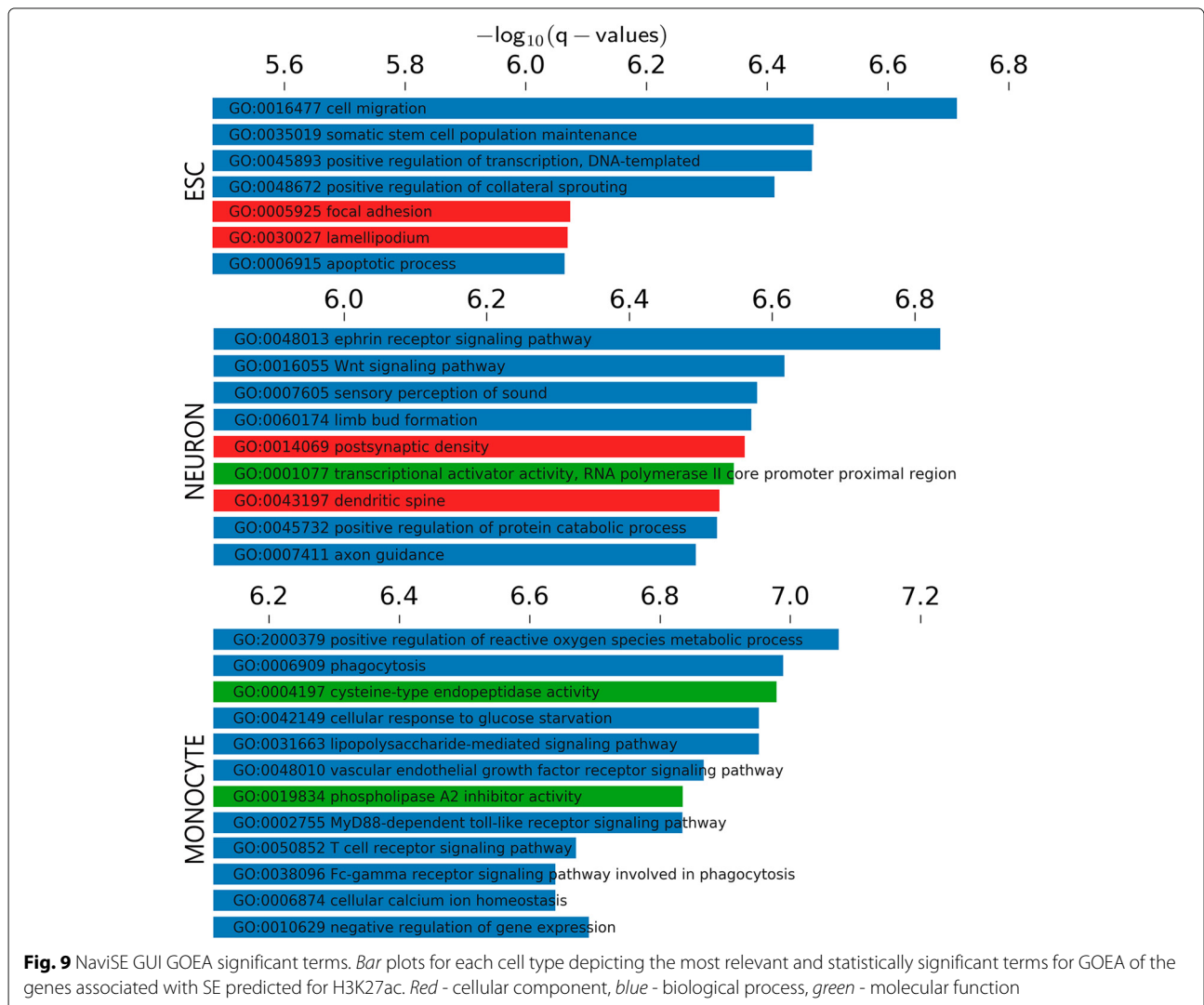
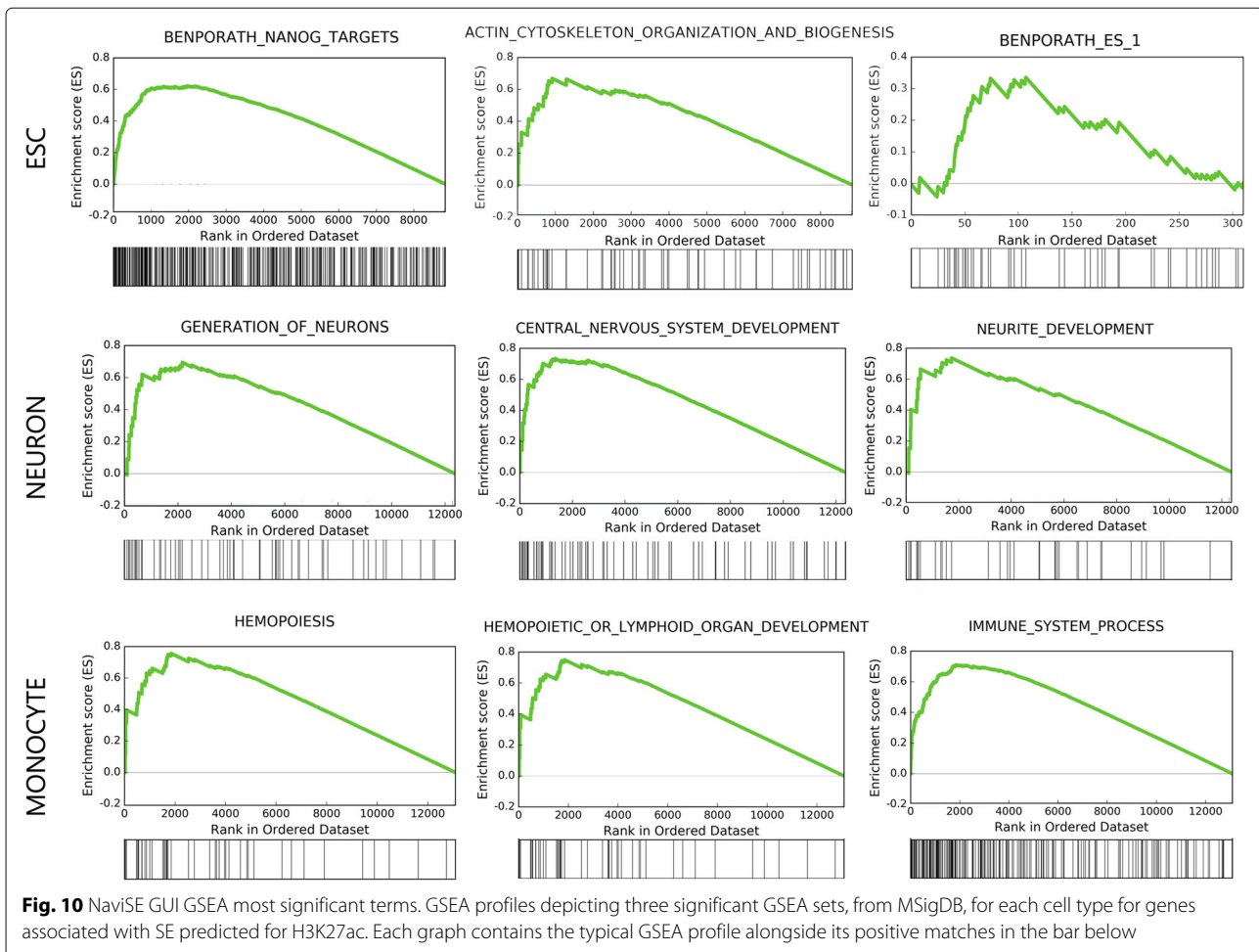


Fig. 9 NaviSE GUI GOEA significant terms. Bar plots for each cell type depicting the most relevant and statistically significant terms for GOEA of the genes associated with SE predicted for H3K27ac. Red - cellular component, blue - biological process, green - molecular function



terms are *NOTCH2* (related to hematopoiesis), *CD14* (one of the main markers of monocytes), *TLR2* (Toll-like receptor 2, which plays a fundamental role in pathogen recognition and activation of innate immunity [30]), *MAPK13* (is activated by proinflammatory cytokines and cellular stress [31]) or *LYN* (might be involved in the regulation of mast cell degranulation, and erythroid differentiation [32, 33]). Interestingly, *NOTCH1* gene, which is essential for hematopoiesis [34], does not appear in the list of SEs predicted by NaviSE for this dataset.

Enrichr analysis

We performed an Enrichr analysis in order to search genes involved in cellular processes related to each cell type. Most of the found genes, if not mentioned previously, appeared in GSEA and GOEA as well.

For ESC, the Enrichr Reactome presents several terms such as *transcriptional regulation of pluripotent stem cells*; and *POU5F1*, *SOX2*, *NANOG genes related to proliferation*, widely related to embryogenesis. Predominant genes are *FGF2* (implicated in a multitude of physiologic

and pathologic processes, including limb development, angiogenesis, wound healing, and tumour growth [35]), *SOX2* or *NANOG* (TF belonging to Homeobox proteins, critically involved with self-renewal of undifferentiated ESCs, which is also one of Thomson's reprogramming factors [36]). ENCODE and Chromatin Enrichment Analysis (ChEA) TFs includes TFs related to pluripotency (*TCF3*, *NANOG*, *SOX2*, *POU5F1* and *KLF4* as the most relevant) which share several genes, such as *ZMYDN8*, or *DIDO1* (involved in apoptosis, autophagy, and meiosis). Interestingly, and as described by Hnisz et al. [4], we found that the SEs predicted by NaviSE are capable of disclosing a crosstalk between TFs (for instance, all the aforementioned TFs interact with *SOX2* and *NANOG*, according to ENCODE).

As for NEU, Reactome includes significant terms such as *axon guidance* or *semaphorin interactions*, with genes such as *TRIO* or *CDK5R1*; which also appear as genes associated with several TFs such as *REST* (transcriptional repressor that represses neuron-specific genes, such as type II sodium channel gene [37, 38]), determined by ENCODE or TRANSFAC. A gene predicted

to associate with *REST* is *SOX1*, a known neuronal marker.

Regarding MON, Reactome presents several terms such as *immune system*, *innate immune system*, *hemostasis* or *toll-like receptor 2 cascade*, widely related to monocytes, whose associated genes are *TLR2*, *FOS* (implicated as regulator of cell proliferation, differentiation, and transformation, associated with B lymphocyte differentiation and involved in lipopolysaccharide and low density lipoprotein response [39–41]) or *CD86*, expressed by antigen-presenting cells. Binding of this protein to CD28 antigen is a co-stimulatory signal for activation of the T-cell. TRANSFAC and ENCODE include genes associated with TFs like *GATA1*, *GATA2*, *SPI1* or *RUNX1*, among which are *IKZF1* or *JARID2*. Enrichr also determined markers for monocytes or lymphoid cells, such as *RIN3*, *CXCR4*, *TREM1* or *ETV6*.

NaviSE epigenomics signal algebra is able to predict SEs with sharper signals

To evaluate to which extent the use of the epigenomics algebra improves the SE predictions, we have selected combinations of activation and repression epigenetic signals and compared SE predictions of HOMER, ROSE and NaviSE in ESCs. We denote the set formed by a SE software predictor {HOMER, ROSE, NaviSE}, and the set of SEs and TEs derived from an algebra of single or combined epigenetic signals {H3K27ac, H3K4me1, H3K4me3, H3K27ac NOT H3K4me3, H3K27ac NOT H3K27me3, H3K27ac + H3K4me1 - H3K4me3, H3K27ac + H3K4me1 - H3K27me3} as $STIT_{pred-algebra}$. To quantify the results of the different $STIT_{pred-algebra}$, we collected a set of ESC core pluripotency markers from the literature [42] and built a metric of the global goodness of the $STIT_{pred-algebra}$ based on the SE ranking generated for each $STIT_{pred-algebra}$ over the set of ESC markers. As each $STIT_{pred-algebra}$ contains a different number of SEs (thus, producing ranks of different length), to make the different SE ranks comparable, we designed a transformation to re-scale each SE rank, r , given by Eq. 2 into a scaled rank $s(r)$ as follows:

$$s(r) = \frac{r}{|STIT_{pred-algebra}|} \cdot 100 \quad (6)$$

where $|STIT_{pred-algebra}|$ is the number of SEs predicted by each $STIT_{pred-algebra}$. Thus, when we apply Eq. 6 to scale the rank r , it produces a $s(r)$ in the range [0, 100] if the epigenomics signal algebra is predicted as a SE, and $s(r) > 100$ if the signal algebra is predicted as a TE or is not predicted at all. Better 1 $STIT_{pred-algebra}$ assigns lower $s(r)$ s to the SEs associated to ESC gene markers.

To quantify the global performance of each $STIT_{pred-algebra}$, we calculated the average \bar{s} of $s(r)$ over the list of all ESC markers. Therefore, the best

$STIT_{pred-algebra}$ will produce the lowest \bar{s} . We depict the $s(r)$ for the list of ESC gene markers and the list of $STIT_{pred-algebra}$ in the heatmap of Fig. 11a.

We observe three main patterns of behaviour, a group I of genes (from *MED14* until *MYH9*) that has associated a majority of SEs predicted by almost all the $STIT_{pred-algebra}$, some of them not by HOMER, a group II (from *TPD52* until *LRRC2*) that has associated TEs predicted by ROSE and NaviSE $STIT_{pred-algebra}$ but not by HOMER, and a group III (from *RBM14* until *KLF2*) that has associated lower ranked TEs from some of the combined algebras of NaviSE.

Interestingly, no $STIT_{pred-algebra}$ predicts SEs associated with the master regulator of pluripotency *POU5F1/OCT4* (they appear as TEs with H3K4me3 and H3K27ac + H3K4me1 - H3K27me3 from NaviSE), suggesting that *POU5F1* has a subtle epigenomic regulation that hinders the discovery for upstream *POU5F1* regulators, as it has been observed in the computational attempts with unconstrained discovery algorithms to find *ab initio* motifs regulating the *POU5F1* promoter [43].

The plot in Fig. 11b depicts the normalised metric of global performance \bar{s} of each $STIT_{pred-algebra}$, where the lowest values are associated to the best performance. We observe that HOMER-based predictors show the worse performance, NaviSE single epigenomic signal SE predictions are better than those of HOMER and ROSE, and NaviSE H3K27ac + H3K4me1 - H3K4me3 algebra is better than any other single epigenomic signal SE predictions, thus showing the advantage of using the NaviSE epigenomic signal algebra to perform SE predictions.

To illustrate how the profiles of the combined epigenetic signal algebras are developed, we selected the best performing algebra (H3K27ac+H3K4me1-H3K4me3) and depicted its resulting combination and component signals profiles H3K27ac, H3K27me1, H3K4me3 for *NANOG*, (Fig. 11c) and *FOXO1* (Fig. 11d). In both cases, the deletion of the H3K4me3 promoter signal upstream and over the first exon and intron shortens the SE support to focus the SE support upstream of these genes.

Therefore, although there might not be a 'gold standard' on what a real SE is, we can conclude that the SE predictions of NaviSE are better than other predictors', with the added advantage to be fast obtained, fully automatized and comprehensively annotated.

Conclusions

We designed NaviSE to perform automatic parallelised SE prediction from genome-wide epigenetic signals, or an algebra of them, due to an optimization that reduces the necessity of inputting most of the parameters, providing a comprehensive annotation of SEs. NaviSE SE annotation runs from the motifs of TFBSs enriched in SEs through

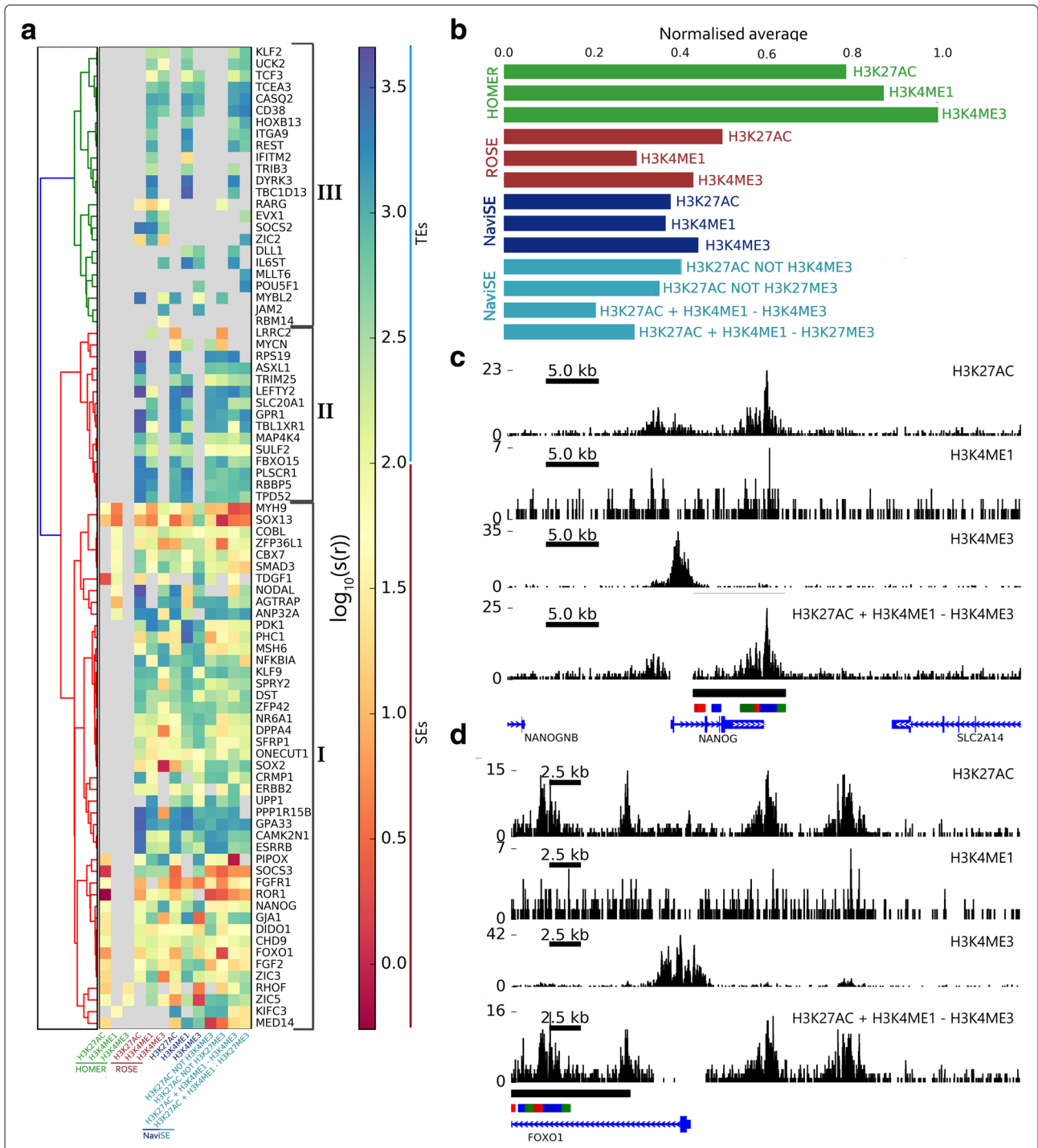


Fig. 11 Performance of the epigenomics algebra on ESC gene markers. **a** Heatmap of the scaled ranking $s(r)$ calculated by Eq. 6 for the SEs predicted by different $STIT_{pred-algebra}$ for ESC gene markers. The scaled ranking is colour coded in red for good ranked (low ranking values) SEs, in yellow (from 2.0 onwards) for good ranked TEs, in green and blue for bad ranked TEs, and in grey for TEs without signal prediction. **b** Global goodness of each $STIT_{pred-algebra}$ over the whole set of ESC gene markers; normalised to the predictor of highest average (HOMER H3K4me3). Epigenomic algebra and single epigenomic signal box plot of peak distribution, depicted by our GVT module, for the SE associated to *NANOG* (**c**) and *FOXO1* (**d**). The bottom row contains the combination of epigenomic signals, and the rows above contain the original single signals. Black lines below the signal represent the SE supports at each sample, and bars in alternating colours below SE bar show the supports of the SE subpeak composition

functional analysis (GOEA, GSEA and enriched metabolic pathways) to PPI networks to a broad tissue prediction, thus, covering a wide range of valuable information. Such integrated annotation is of paramount importance due to the regulatory nature of the SEs, which have been described as key players in the determination of cell fate and in the involvement in the mechanisms of disease. Simultaneously, NaviSE performs all these tasks optimizing the use of the computer resources, identifying the available cores and main memory, and takes maximum advantage of them in function of the task requirements.

Furthermore, the automatic recognition of multiple file formats and the capability of working with replicates and controls, alongside with the possibility of integrating onto other pipelines or running multiple samples with multiple replicates and signal algebras at once with a simple script in Python, makes NaviSE a foremost tool for an efficient study of SEs. Due to all these capabilities, NaviSE is a time-saving and user-friendly tool for SE analysis.

To validate the biological performance of NaviSE, we applied it to predict the SEs on real data sets of several cell types with a different level of differentiation and commitment, and predicted in all cases SE-associated genes in agreement with the expected cell-specific markers. In the case of ESCs, NaviSE predicted SEs on the ESC markers *NANOG* and *SOX2*, in the case of neurons it predicted the *SOX1* and *CDK5R1* neuron markers, and in the case of monocytes, predicted the *CD86* and *CXCR4* monocyte markers.

The Additional file 1 provides a complete guide to the software installation and use instructions.

Availability and requirements

NaviSE. Project name: NaviSE. NaviSE is freely available at <https://sourceforge.net/projects/navise-superenhancer/>.

Operating system: Linux 64bit (Ubuntu 11.04).

Programming language: Python 3.5. License: GNU GPL.

Additional file

Additional file 1: Supplementary Information. Manual for installation, use and running examples of NaviSE. (pdf 33792 kb)

Abbreviations

ChIP-Seq: Chromatin immunoprecipitation sequencing; ChEA: ChIP enrichment analysis; DAG: Directed acyclic graph; ESC: Embryonic stem cell; FDR: False discovery rate; GB: Gigabyte; GO: Gene ontology; GSEA: Gene Set enrichment analysis; GOEA: Gene ontology enrichment analysis; GUI: Graphic user interface; GVT: Genomic viewer tool; HOMER: Hypergeometric optimization of motif enrichment; MACS: Model-based Analysis for ChIP-Seq; MSigDB: Molecular signatures database; MON: Monocyte; NEU: Neuron; NGS: Next generation sequencing; P3BSseq: Parallel processing pipeline software for automatic analysis of bisulfite sequencing data; PPI: Protein-protein interaction; ROSE: Ranking of superenhancers; SE: Superenhancer; TE: Typical enhancer; TF: Transcription factor; TFBS: Transcription factor binding site; TSS: Transcription start site

Acknowledgments

We thank Iñigo García de las Cuevas for providing the first model of the pipeline for file processing.

Funding

This work was supported by grants from the Ministry of Economy and Competitiveness, Spain, MINECO grants P116/01430 and BFU 2016-7798-P; DFG10/15, DFG15/15 and DFG141/16 from Diputación Foral de Gipuzkoa, Spain, and grant I114/00016 from I+D+I National Plan 2013–2016 of Carlos III Health Institute, Spain, FEDER funds and IKERBASQUE, Basque Foundation for Science, Spain; who have provided funding for the computational infrastructure to develop the project, of which this software is a part. AMA received a fellowship from the Ikerbasque program (IkaC_2016_1_0017) of the Department of Education of the Basque Government, to support the development of the software as part of his final degree project. None of the funding bodies have played any part in the design of the study, in the collection, analysis, and interpretation of the data, or in the writing of the manuscript.

Authors' contributions

AMA developed and tested the software, obtained the results from biological samples, and wrote the manuscript. MAE contributed to the development of the installation software and the initial pipeline. AI designed and supervised the project. MJAB designed and supervised the project, and wrote the manuscript. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Consent for publication

Not applicable.

Ethics approval and consent to participate

Not applicable.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details

¹Computational Biology and Systems Biomedicine, Biodonostia Health Research Institute, 20014 San Sebastián, Spain. ²Tissue Engineering Laboratory, Bioengineering Area, Biodonostia Health Research Institute, 20014 San Sebastián, Spain. ³Department of Biochemistry and Molecular Biology, University of the Basque Country, 48940 Leioa, Spain. ⁴IKERBASQUE, Basque Foundation for Science, 48013 Bilbao, Spain.

Received: 7 December 2016 Accepted: 18 May 2017

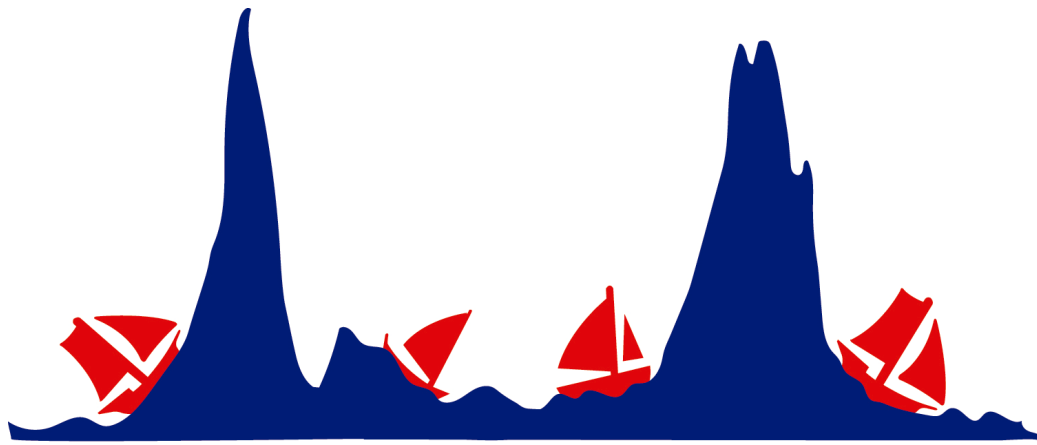
Published online: 06 June 2017

References

- Pott S, Lieb JD. What are super-enhancers? *Nature Gen.* 2014;47(1):8–12.
- Whyte WA, Orlando DA, Hnisz D, Abraham BJ, Lin CY, Kagey MH, Rahl PB, Lee TI, Young RA. Master Transcription Factors and Mediator Establish Super-Enhancers at Key Cell Identity Genes. *Cell.* 2013;153:307–19.
- Adam RC, Yang H, Rockowitz S, Larsen SB, Nikolova M, Oristian DS, Polak L, Kadaja M, Asare A, Zheng D, Fuchs E. Pioneer factors govern super-enhancer dynamics in stem cell plasticity and lineage choice. *Nature.* 2015;521(7552):366–70.
- Hnisz D, Abraham BJ, Lee TI, Lau A, Saint-André V, Sigova AA, Hoke HA, Young RA. Super-Enhancers in the Control of Cell Identity and Disease. *Cell.* 2013;155:934–47.
- Heyn H, Vidal E, Ferreira HJ, Vizoso M, Sayols S, Gomez A, Moran S, Boque-Sastre R, Guil S, Martinez-Cardus A, Lin CY, Royo R, Sanchez-Mut JV, Martinez R, Gut M, Torrents D, Orozco M, Gut I, Young RA, Esteller M. Epigenomic analysis detects aberrant super-enhancer DNA methylation in human cancer. *Genome Biol.* 2016;17(11):1–16.
- Hnisz D, Schuijers J, Lin CY, Weintraub AS, Abraham BJ, Lee TI, Bradner JE, Young RA. Convergence of developmental and oncogenic signaling pathways at transcriptional super-enhancers. *Mol Cell.* 2015;58(2):362–70.

7. Lovén J, Hoke HA, Lin CY, Lau A, Orlando DA, Vakoc CR, Bradner JE, Lee TI, Young RA. Selective Inhibition of Tumor Oncogenes by Disruption of Super-Enhancers. *Cell*. 2013;153(2):320–34.
8. Luu PL, Gerovska D, Arrospide-Elgarresta M, Retegi-Carrión S, Schöler HR, Araúz-Bravo MJ. P3BSSEQ: Parallel processing pipeline software for automatic analysis of bisulfite sequencing data. *Bioinformatics*. 2017;33(3):428–31.
9. Langmead B, Salzberg SL. Fast gapped-read alignment with Bowtie 2. *Nat Methods*. 2012;9:357–9.
10. Lee WP, Stromberg MP, Ward A, Stewart C, Garrison EP, Marth GT. MOSAIK: A Hash-Based Algorithm for Accurate Next-Generation Sequencing Short-Read Mapping. *PLoS ONE*. 2014;9(3):1–11.
11. Dobin A, Davis CA, Schlesinger F, Drenkow J, Zaleski C, Jha S, Batut P, Chaisson M, Gingeras TR. STAR: ultrafast universal RNA-seq aligner. *Bioinformatics*. 2013;29(1):15–21.
12. Li H, Durbin R. Fast and accurate short read alignment with Burrows-Wheeler Transform. *Bioinformatics*. 2009;25:1754–60.
13. Zhang Y, Liu T, Meyer CA, Eeckhoute J, Johnson DS, Bernstein BE, Nusbaum C, Myers RM, Brown M, Li W, Liu XS. Model-based Analysis of ChIP-Seq (MACS). *Genome Biol*. 2008;9(9):137.
14. Hnisz D, Schuijers J, Lin CY, Weintraub AS, Abraham BJ, Lee TI, Bradner JE, Young RA. Convergence of developmental and oncogenic signaling pathways at transcriptional super-enhancers. *Mol Cell*. 2015;58:1–9.
15. Haibao T, Klopfenstein DV, Pedersen B, Ramirez F, Naldi A, Flick P, Yunes J, Sato K, Mungall C, Stupp G, DeTomaso D, Botvinnik O. GOATOOLS: Tools for Gene Ontology. Zenodo. 10.5281/zenodo.31628.
16. Chen EY, Tan CM, Kou Y, Duan Q, Wang Z, Meirelles GV, Clark NR, Ma'ayan A. Enrichr: interactive and collaborative HTML5 gene list enrichment analysis tool. *BMC Bioinforma*. 2013;14(128):1–14.
17. Szklarczyk D, Franceschini A, Wyder S, Forslund K, Heller D, Huerta-Cepas J, Simonovic M, Roth A, Santos A, Tsafou KP, Kuhn M, Bork P, Jensen LJ, von Mering C. STRING v10: protein-protein interaction networks, integrated over the tree of life. *Nucleic Acids Res*. 2015;43:447–52.
18. Jaïoun K, Teerapabolarn K. An improved binomial approximation for the hypergeometric distribution. *Appl Math Sci*. 2014;8(13):613–7.
19. Subramanian A, Tamayo P, Mootha VK, Mukherjee S, Ebert BL, Gillette MA, Paulovich A, Pomeroy SL, Golub TR, Lander ES, Mesirov JP. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *PNAS*. 2005;102(43):15545–50.
20. Edgar R, Domrachev M, Lash AE. Gene Expression Omnibus: NCB1 gene expression and hybridization array data repository. *Nucleic Acids Res*. 2002;30(1):207–10.
21. Rebhan M, Chalifa-Caspi V, Prilusky J, Lancet D. GeneCards: integrating information about genes, proteins and diseases. *Trends Genet*. 1997;13(4):163.
22. Tyner C, Barber GP, Casper J, Clawson H, Diekhans M, Eisenhart C, Fischer CM, Gibson D, Gonzalez JN, Guruvadoo L, Haeussler M, Heitner S, Hinrichs AS, Karolchik D, Lee BT, Lee CM, Nejad P, Raney BJ, Rosenbloom KR, Speir ML, Villarreal C, Vivian J, Zweig AS, Haussler D, Kuhn RM, Kent WJ. The UCSC Genome Browser database: 2017 update. *Nucleic Acids Res*. 2017;45(4):626–34.
23. Afzal AR, Jeffery S. One gene, two phenotypes: ROR2 mutations in autosomal recessive Robinow syndrome and autosomal dominant brachydactyly type B. *Hum Mutat*. 2003;22(1):1–11.
24. Lim LS, Hong FH, Kurnarso G, Stantonon LW. The pluripotency regulator Zic3 is a direct activator of the Nanog promoter in ESCs. *Stem Cells*. 2010;28(11):1961–9.
25. Takahashi K, Yamanaka S. Induction of Pluripotent Stem Cells from Mouse Embryonic and Adult Fibroblast Cultures by Defined Factors. *Cell*. 2006;126(4):663–76.
26. Luo F, Zhang J, Burke K, Miller RH, Yang Y. The Activators of Cyclin-Dependent Kinase 5 p35 and p39 Are Essential for Oligodendrocyte Maturation, Process Formation, and Myelination. *J Neurosci*. 2016;36(10):3024–37.
27. Kang J, Park H, Kim E. IRSp53/BAIAP2 in dendritic spine development, NMDA receptor regulation, and psychiatric disorders. *Neuropharmacol*. 2016;100:27–39.
28. Castro DS. One more factor joins the plot: Pbx1 regulates differentiation and survival of midbrain dopaminergic neurons. *EMBO J*. 2016;35(18):1957–9.
29. Villaescusa JC, Li B, Toledo EM, Rivetti di Val Cervo P, Yang S, Stott SR, Kaiser K, Islam S, Gyllborg D, Laguna-Goya R, Landreh M, Lönnberg P, Falk A, Bergman T, Barker RA, Linnarsson S, Selleri L, Arenas E. A PBX1 transcriptional network controls dopaminergic neuron development and is impaired in Parkinson's disease. *EMBO J*. 2016;35(18):1963–78.
30. Jin MS, Kim SE, Heo JY, Lee ME, Kim HM, Paik SG, Lee H, Lee JO. Crystal structure of the TLR1-TLR2 heterodimer induced by binding of a tri-acylated lipopeptide. *Cell*. 2007;130:1071–82.
31. Hu MC, Wang YP, Mikhail A, Qiu WR, Tan TH. Murine p38-delta mitogen-activated protein kinase, a developmentally regulated protein kinase that is activated by stress and proinflammatory cytokines. *J Biol Chem*. 1999;274(11):7095–102.
32. Toubiana J, Rossi AL, Belaidouni N, Grimaldi D, Pene F, Chafey P, Comba B, Camoin L, Bismuth G, Claessens YE, Mira JP, Chiche JD. Src-family-tyrosine kinase Lyn is critical for TLR2-mediated NF- κ B activation through the PI 3-kinase signaling pathway. *Innate Immunol*. 2015;21(7):685–97.
33. Parravicini V, Gadina M, Kovarova M, Odom S, Gonzalez-Espinosa C, Furumoto Y, Saitoh S, Samelson LE, O'Shea JJ, Rivera J. Fyn kinase initiates complementary signals required for IgE-dependent mast cell degranulation. *Nat Immunol*. 2002;3(8):741–8.
34. Kumano K, Chiba S, Kunisato A, Sata M, Saito T, Nakagami-Yamaguchi E, Yamaguchi T, Masuda S, Shimizu K, Takahashi T, Ogawa S, Hamada Y, Hirai H. Notch1 but Not Notch2 Is Essential for Generating Hematopoietic Stem Cells from Endothelial Cells. *Immunity*. 2003;18:699–711.
35. Ortega S, Ittmann M, Tsang SH, Ehrlich M, Basilio C. Neuronal defects and delayed wound healing in mice lacking fibroblast growth factor 2. *PNAS*. 1998;95(10):5672–77.
36. Yu J, Vodyanik MA, Smuga-Otto K, Antosiewicz-Bourget J, Frane JL, Tian S, Nie J, Jonsdottir GA, Ruotti V, Stewart R, Slukvin II, Thomson JA. Induced pluripotent stem cell lines derived from human somatic cells. *Science*. 2007;318(5858):1917–20.
37. Chong JA, Tapia-Ramírez J, Kim S, Toledo-Aral JJ, Zheng Y, Boutros MC, Altshuler YM, Frohman MA, Kraner SD, Mandel G. REST: A Mammalian Silencer Protein That Restricts Sodium Channel Gene Expression to Neurons. *Cell*. 1995;80:949–57.
38. Schoenherr CJ, Anderson DJ. The neuron-restrictive silencer factor (NRSF): a coordinate repressor of multiple neuron-specific genes. *Science*. 1995;267(5202):1360–3.
39. Phuchareon J, Tokuhisa T. Deregulated c-Fos/AP-1 accelerates cell cycle progression of B lymphocytes stimulated with lipopolysaccharide. *Immunobiology*. 1995;193(5):391–9.
40. Kang JG, Sung HJ, Jawed SI, Brenneman CL, Rao YN, Sher S, Facio FM, Biesecker LG, Quyyumi AA, Sachdev V, Hwang PM. FOS expression in blood as a LDL-independent marker of statin treatment. *Atherosclerosis*. 2010;212(2):567–70.
41. Ohkubo Y, Arima M, Arguni E, Okada S, Yamashita K, Asari S, Obata S, Sakamoto A, Hatano M, O-Wang J, Ebara M, Saisho H, Tokuhisa T. A role for c-fos/activator protein 1 in B lymphocyte terminal differentiation. *J Immunol*. 2005;174(12):7703–10.
42. Kim J, Woo AJ, Chu J, Snow JW, Fujiwara Y, Kim CG, Cantor AB, Orkin SH. A Myc Network Accounts for Similarities between Embryonic Stem and Cancer Cell Transcription Programs. *Cell*. 2010;143(2):313–24.
43. Müller-Molina AJ, Schöler HR, Araúz-Bravo MJ. Comprehensive human transcription factor binding site map for combinatorial binding motifs discovery. *PLoS One*. 2012;7(11):49086.

8 APPENDIX III: NAVISE INSTALLATION AND USE MANUAL



NaviSE

NaviSE: Superenhancer Navigator

Documentation

Version: 1.1

Alex M. Ascensión and Marcos J. Araúzo-Bravo

2017

Contents

What is NaviSE?	3
Installation	4
Step-by-step installation	4
Basic linux programs	7
Installing SRA-TO-BAM programs	8
Gene Ontology	8
Installing Goatools (Gene Ontology)	8
Installing Genome viewer associated files	9
Installing HOMER	9
Installing GSEA and beautifulsoup	10
Installing programs for Enrichr/StringDB data extraction	10
Last but not least... setting NaviSE path	12
Automatic installation	12
NaviSE Genomes	13
Installation of other genomes	13
Commands	17
Running NaviSE	23
Parallelization of NaviSE	28
NaviSE output	29
Main page	30
SuperEnhancer table	30
NaviSE Graphs	32
GOEA results	34
HOMER analysis	38
Enrichr results	40
StringDB results	41
GSEA results	41

What is NaviSE?

NaviSE (SuperEnhancer Navigator) is a software designed to obtain analytic superenhancer (SE) data from ChIP-seq, DNA-seq, ATAC-seq or similar data. NaviSE executes a series of commands which extract information from raw data and include information about associated genes, overrepresented motifs or Gene Ontology Enrichment Analysis (GOEA). All the information is gathered, processed and exported to an html file that the user can navigate and extract the relevant information from for their analysis.

NaviSE acts *mainly* as a 'program of programs', i.e. uses different software from third parties to process raw data, analyse motifs, perform GOEA or Gene Set Enrichment Analysis (GSEA). It also uses information from web databases (Jaspar, Transfac, String, etc.) which complement the original information. Finally, NaviSE also includes self-processed information, such as graphs or plots, which users may find supportive for the analysis of their results.

In other words, the only requirements for NaviSE to work are data from ChIP-seq experiments (data formats described in [data formats](#)) and introducing the [commands](#) on the console. All the results are presented in an html report (explained in [NaviSE output](#)) for the sake of simplicity and handiness, which allows the user to navigate through the information, and access complementary information via links.

Installation

As explained before, NaviSE works with different third-party software, so before your first NaviSE run, you will need to spend some time to install all these modules.

In order to download and install the files, download the original files from <https://sourceforge.net/projects/navise-superenhancer/>. In this page three different files appear: (1) **Python files** - It contains all the files NaviSE requires for proper working of the program, (2) **Programs** - It contains some installation files for third-party software, and (3) **Files** - It contains files which NaviSE requires for proper working.

First of all, download all the files and extract them into a directory where all the compressed files will be extracted. We recommend to create a general directory such as *Programs* where both NaviSE and other related programs will be located, and then extract the files into a subdirectory. In order to extract the files, right click on each compressed file and press on *Extract here* or any other similar message (this requires a uncompressing software to be installed). In total, two subdirectories (*Programs* and *Files*), along with several *.py* files should appear in the directory.

The installation steps appear on the Fig. 1.

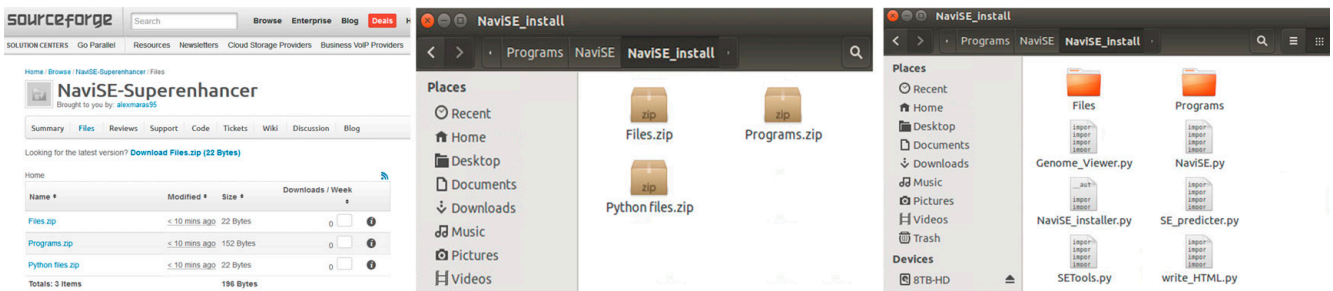


Figure 1: NaviSE binaries download.

Warning: Please, DO NOT move any of the files into other directories, nor delete any files unless they are dispensable.

Now that all the files have been downloaded, you can proceed to install NaviSE and its dependencies step by step or automatically.

Step-by-step installation

NaviSE is developed in **python 3.5**, so a python distribution is required.

Warning: Python2.7 is not allowed by NaviSE, I will never downgrade my program. Also, there is a newer python distribution, python3.6, but it still has incompatibilities with some modules, so we have to stick to python3.5.

Linux should come with its own python distribution, although it doesn't implement some modules

required for NaviSE. Therefore, we must install a Python distribution which includes these modules (pandas, numpy, etc.). Our election is [anaconda distribution](#), which includes essential packages like Numpy or Scipy, used throughout NaviSE run; and also installs dependencies related to packages when a module is installed, so manual installation of the dependencies is not required (which usually are not version wise correlated, and NaviSE may crash). Moreover, it can be used in other projects as well.

Warning: We expect users to install anaconda. If other installation lines are followed or other distributions are installed, NaviSE will not work, as it requires anaconda to set paths to he programs. We apologize for the inconveniences.

From now on, we will work via Linux command prompt (terminal), executing a series of commands. Those commands will appear in verbatim mode (like this text), for easier interpretation. The terminal can be prompted pressing `Ctrl` + `Alt` + `T`, which should appear like a black or violet window, in this fashion (Fig. 2):

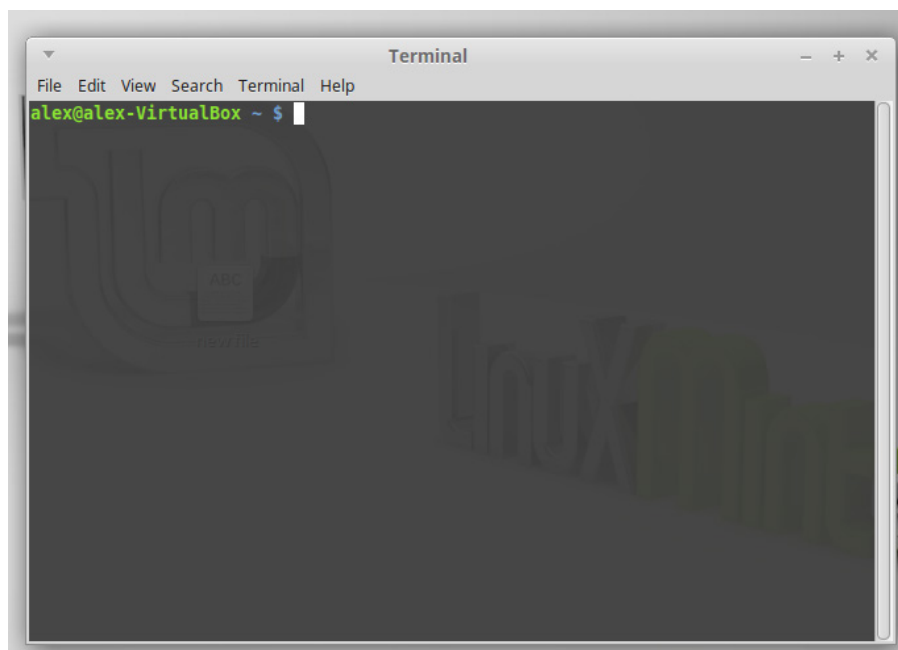


Figure 2: Basic prompt.

The first step is to download anaconda (version 3.4 in this case). It can be downloaded in two ways:

- By downloading the file (with extension `.sh`) from the [official page](#).
- By downloading it via the command

```
wget https://repo.continuum.io/archive/Anaconda3-4.1.1-Linux-x86_64.sh
```

In both cases, downloaded file with a name similar to *Anaconda3-4.X.X-Linux-x86_64.sh* should appear. Now, we have to run the installation file, for which the easiest way to do is to write in the terminal bash, hit the `Space bar`, and then grab the downloaded file and drag it the terminal. A screen like this should appear (Fig. 3):

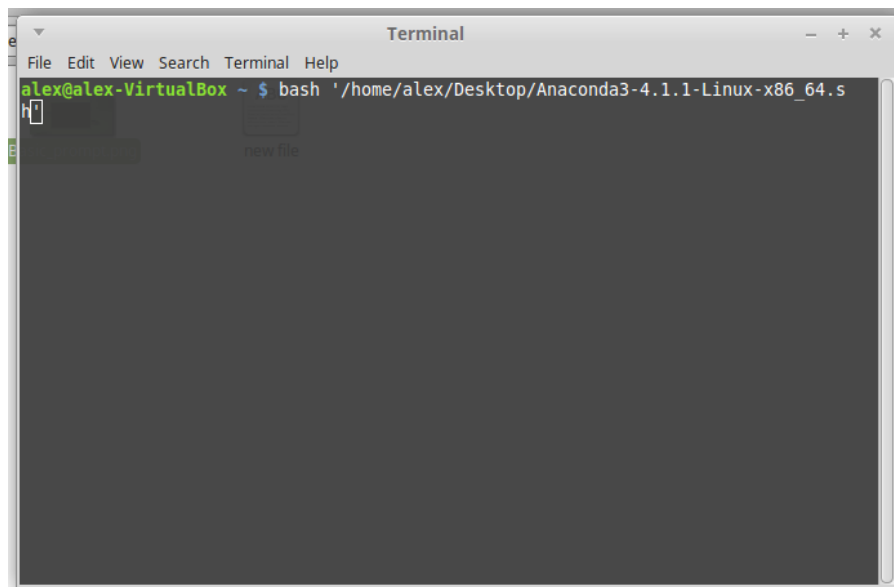


Figure 3: Anaconda download.

Hit `Enter` and wait. Several messages may appear, being the first that if you want to relocate your installation directory (the standard directory is the */home* directory). You can leave it as it is or write something like */YOURUSERNAME/home/Programs/anaconda3*, as we will install other programs and it is a good practice to keep everything in the same directory.

Finally, once everything is installed, you will be asked about setting a PATH file; write *Y* and hit `Enter`. Remember NOT to change the directory of anaconda once installed, as it will not work. If so, you would need to change your path file of anaconda (described later).

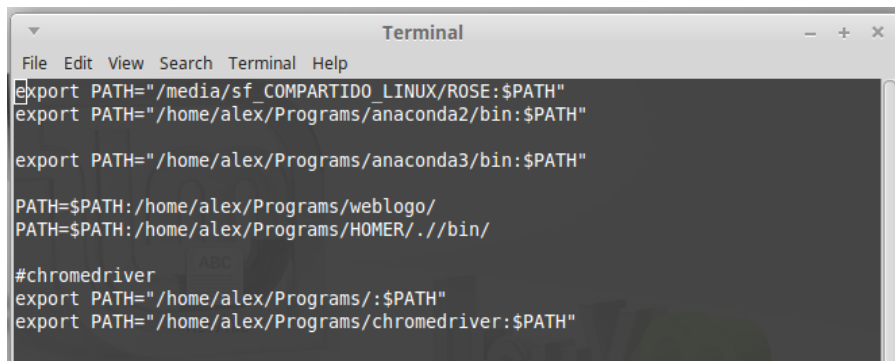
Setting a path to a program

With the last changes I performed, it is not fully necessary to set the path to anaconda or HOMER, because NaviSE automatically recognizes the main files from the programs. However, it is highly recommendable to set the paths to those programs (NaviSE installer automatically does it) in case they will be used independently with other software on their own.

Setting a path means changing a file which tells Linux where the executables of the program are located. If the PATH is absent, changed or the file is relocated, Linux will not be able to detect the location of this program, and when running it, it will be considered as 'non-existent'. On default, anaconda remembers the installation site and adds the PATH automatically. However, if you need to

change the location of anaconda or you have not selected *Yes* when installing anaconda, the PATH must be set for proper recognition. Setting a PATH requires the following steps:

- Open the terminal and write `vi ~/.bashrc`. *vi* is a text editor and */.bashrc* is the file with the location of the paths. If it asks to create a new file, or that the file exists and you want to modify it, write *Y* and hit `Enter`. An empty window or a window like that should appear (Fig. 4):



```
Terminal
File Edit View Search Terminal Help
export PATH="/media/sf_COMPARTIDO_LINUX/ROSE:$PATH"
export PATH="/home/alex/Programs/anaconda2/bin:$PATH"

export PATH="/home/alex/Programs/anaconda3/bin:$PATH"

PATH=$PATH:/home/alex/Programs/weblogo/
PATH=$PATH:/home/alex/Programs/HOMER/./bin/

#chromedriver
export PATH="/home/alex/Programs/:$PATH"
export PATH="/home/alex/Programs/chromedriver:$PATH"
```

Figure 4: Vi editor and path setting.

- Enter into the edition mode by pressing `i`.
- Now, write the following line: `export PATH="XXX/anaconda3/bin:$PATH"` where *XXX* is the path of the directory where anaconda is located.
- Once the line is written, press `Esc` and write `:wq` to save the changes and exit.
- In order for the changes to apply, close the terminal and open it again. To reassure that the path has been correctly set, write `which anaconda`, if the output is `/XXX/anaconda3/bin/anaconda/` (*XXX* being the installation dir), the path is correctly set; if nothing appears, make sure the path is correctly written or that the command prompt has been restarted.

Basic linux programs

Before beginning with the installation of NaviSE components, we will need to install python2.7 and pip-2.7 to install some basic components. First, we check the presence of both programs.

For python2.7, write `which python2.7` and for pip write `which pip2.7`. If in any of them a path appears, it means that the program has been installed. If nothing appears, it means that the program is not installed still.

In order to install the programs:

- For python2.7, write `sudo apt-get install python2.7`. This process may require writing a password.

- For pip2.7, it will be installed in two steps:
 - 1) `wget https://bootstrap.pypa.io/get-pip.py`
 - 2) `sudo python2.7 get-pip.py`

Installing SRA-TO-BAM programs

For the following programs, just write each command in the prompt, press and write Y when asked:

- MACS: `pip2.7 install macs2`
- Sra-Tools: `conda install sra-tools`
- Samtools: `conda install samtools`
- Bedtools: `conda install bedtools`
- FastQC: `conda install fastqc`

NaviSE integrates several read aligners. Bowtie2 is the aligner by defect, and STAR, BWA, and MOSAIK are other aligners which can be used to transform .fastq files into .bam or .sam. The installation of the aligners is as follows:

- Bowtie2: `conda install bowtie2`
- STAR: `conda install c bioconda star`
- MOSAIK: `conda install c bioconda mosaik`
- BWA: `conda install -c judowill bwa`

Gene Ontology

Gene Ontology file binaries (installation files) are located in the Programs directory of NaviSE files. The location of this file must be indicated when running NaviSE, so it is important to locate it in a known place (for instance, in the same directory where anaconda is installed).

Please, mind not to rename, cut or delete any file inside *Programs* directory. If so, NaviSE may crash in the middle of the run.

Installing Goatools (Gene Ontology)

Goatools requires both an installation and some minor fixes that are patched in a file that comes in *Programs* directory. In order to install Goatools, follow these steps.

- First, install goatools by typing `pip install goatools==0.6.5` in the command prompt.
- We will also install `wget` `conda install wget` in the command prompt.
- Now, we have to apply the patch. Locate a file named `goatools` in the *Gene_Ontology* directory. Copy this file and paste it in your anaconda installation directory: `XXX/anaconda3/lib/python3.5/site-packages`. It will ask if you want to replace the file, say Yes.

Goatools works with third-party software, which is required as well:

- Install pyparsing by typing `easy_install pyparsing` in the command prompt.
- Install fisher by typing `easy_install fisher` in the command prompt.
- Then, install graphviz by typing `pip install graphviz` in the command prompt.
- Finally, install pydot2 typing `pip install pydot2`. pydot2 also requires to be patched, for which you only have to copy the `pydot.py` located in the *Gene_Ontology* directory, and paste it in `XXX/anaconda3/lib/python3.5/site-packages/` (say Yes if Linux ask you whether you want to replace the file).

Installing Genome viewer associated files

The programs required by the genome viewer to run are:

- pysam: `conda install -c bioconda pysam`
- pysamstats: `conda install -c bioconda pysamstats`

Installing HOMER

In order to run HOMER, two third-party software are required:

- Install weblogo by typing `conda install -c percyfall weblogo`
- Then, blatz by typing `conda install -c bioconda blat`

Now for HOMER installation, follow these steps:

In order to install HOMER, download this [file](#) from HOMER website and place it in the HOMER installation directory (make a directory named *HOMER* inside the directory where anaconda is located, for instance). Then, open the terminal and write `perl` and press ; drag the downloaded `configureHomer.pl` file into the prompt (the route to the file should appear), press and write `install`. The terminal should look like this (Fig. 5):

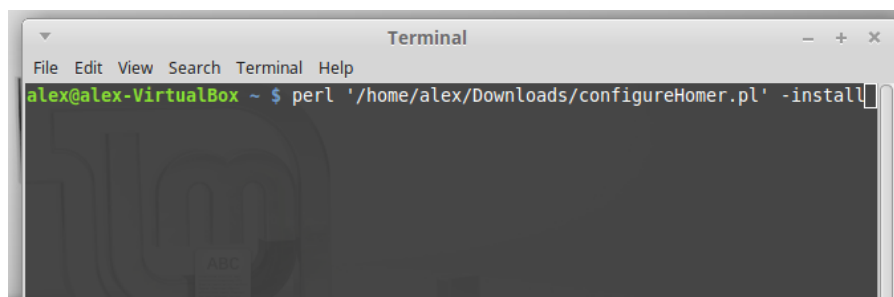


Figure 5: Homer installation.

```
double max = 1-clusters[0].distance;
^
Software Installed. If not done so already, add the homer programs to y
our executable path.
Add this line to your .bash_profile or .bashrc file (or other depending
on your shell):
PATH=$PATH:/home/alex/Downloads//bin/
Simply typing "findMotifs.pl" should work before running Homer.
alex@alex-VirtualBox ~ $
```

Figure 6: Homer PATH setting.

During the installation HOMER, will check whether all the third-party software was installed. If so, the installation continues (otherwise a message appears and HOMER waits 10 seconds for the user's response) and a message like that appears when the installation is finished (Fig. 6):

The message that appears in the red-squared area contains a path similar to the anaconda installation one, that has to be added to the PATH file (see [Setting a path to a program](#)), adding in into a new line.

Following the installation, we need to load information about the genome and the promoters of the animal. In order to install information about the genome, we follow the same steps used for homer installation and add the *name* of the genome. For instance, if we were interested in the version *hg38* from human, we should write:

```
perl PATH-TO-ConfigureHomer.pl -install hg38
```

HOMER should be able to recognize the genome and will install the information. This process may take some minutes, so it is recommendable to install other programs on the meantime.

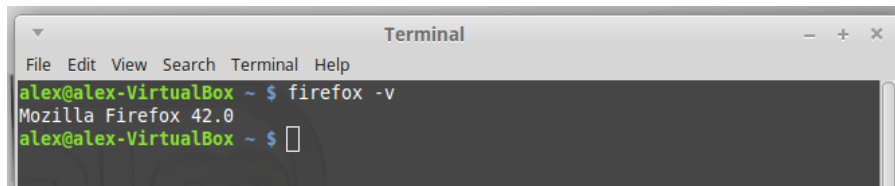
Installing GSEA and beautifulsoup

For those two programs, the installation is quite simple: write these two lines at the command prompt:

- For GSEA: `conda install -c bioconda gseapy`
- For beautifulsoup: `conda install -c anaconda beautifulsoup4=4.5.1`

Installing programs for Enrichr/StringDB data extraction

In order to extract information from Enrichr and String, we need to have a version of Firefox \leq 45.0.2. In order to check your firefox version write `firefox -v` in the prompt. The version should appear like this (Fig. 7):



```
Terminal
File Edit View Search Terminal Help
alex@alex-VirtualBox ~ $ firefox -v
Mozilla Firefox 42.0
alex@alex-VirtualBox ~ $
```

Figure 7: Firefox version.

Downgrading Firefox

If you need to downgrade firefox, follow these steps.

Warning: despite these steps are necessary (selenium does not work with Firefox versions more recent than 45) the following steps will uninstall your current Firefox, so make a security copy of all your bookmarks or history if necessary. Moreover, Firefox does not 'really' support downgrades, so it is possible that the new installation will induce some *errors*, like no icon or no direct access to firefox on program bar or Desktop (although you can just run Firefox typing 'firefox' in the command prompt).

We apologize for the inconveniences. Some solutions we propose on the meantime are installing [chromium](#) and use it as the default browser, or just reinstall firefox from the [official page](#) when you are done with the analysis. It is not *that* hard to do the downgrade when necessary anyways.

Note: theoretically, not installing the software for this part should not induce any errors (it will just not appear on the report), although the information obtained with this part is interesting and, therefore, worth the try.

- First, uninstall Firefox: `sudo apt-get remove firefox`
- Download the v45 release of Firefox:
 - For 32-bit systems:

```
wget http://ftp.mozilla.org/pub/mozilla.org/firefox/releases/45.0/linux-i686/en-US/firefox-45.0.tar.bz2
```
 - For 64-bit systems (common):

```
wget http://ftp.mozilla.org/pub/mozilla.org/firefox/releases/45.0/linux-x86_64/en-US/firefox-45.0.tar.bz2
```
- Extract the downloaded file: `tar xvjf firefox-45.0.tar.bz2`
- We move the file onto a *bridge* directory: `sudo mv firefox/ /opt/firefox3`
- We backup our current Firefox distribution: `sudo mv /usr/bin/firefox /usr/bin/firefox-old`
- Configure firefox: `sudo ln -s /opt/firefox3/firefox /usr/bin/firefox`

Now that firefox has been downgraded the following programs are required. As the previous programs, just type each line into the command prompt:

- Selenium: `pip install selenium`
- Selenium libraries: `sudo pip install -U selenium`
- pyvirtualdisplay (it allows to run the programs in the background): `pip install virtualdisplay`
- Xvft (required by pyvirtualdisplay): `sudo apt-get install xvft python-pip`

Last but not least... setting NaviSE path

Although this last step is not necessary, we highly recommend doing it. By setting the path to NaviSE, in order to run NaviSE in subsequent times, it will only be required to write `python3.5 NaviSE.py` in the command prompt, instead of the full path of *NaviSE.py* location.

In order to set the path, the following line must be added to the `bashrc` file (see how to do it in [Setting a path to a program](#)):

```
PATH=PATH$:XXX
```

Where XXX is the directory where *NaviSE.py* file is located.

Automatic installation

We have recently developed a python file which allows easy NaviSE installation without going through the tedious step-by-step installation. In order to install NaviSE automatically, head to the directory of NaviSE and look for *NaviSE_installer.py* file. Then, open the command prompt (pressing `Ctrl` + `Alt` + `T`) and write `sudo python3 PATH-TO-FILE/NaviSE_installer.py -r XX -c YY -d ZZ`, which will load installation program.

The `-r`, `-c` and `-d` are flags which indicate the location of installation files, as follows:

- `-r`: This flag refers to the directory where files will be installed. If conda or HOMER have not been previously installed, NaviSE will use this path as the main path to install conda and HOMER. If they have already been installed, NaviSE will use this path to install other packages. This flag is REQUIRED.
- `-c`: If conda has already been installed, include the directory of anaconda in this flag. NaviSE will verify the version of Python installed and, if something fails, conda will be installed at the directory mentioned in `-r`. This flag is OPTIONAL.
- `-d`: similar to `-c`, if HOMER has been installed, NaviSE will check that HOMER is fine and that the basic genomes have already been installed. If the genomes are missing, NaviSE will install them at that directory. If something else fails, NaviSE will install HOMER at the directory mentioned in `-r`. This flag is OPTIONAL.

If the program does not run, it may yield one error:

- Python3 is not installed in the system. You can install Python3 by typing in the console `sudo apt-get install python3`

In that case, the console might ask you to input a password. After you do it, all the basic programs will be installed and you will be able to run the installation program.

The installation may take a while (even more than an hour for some computers), so we recommend to read the manual thoroughly in the meantime to understand how to run NaviSE properly.

Warning: this installer checks the presence of HOMER and Anaconda installations, so as not to install them in other places. If anaconda is already installed, please, USE `-c FLAG` to indicate the installation directory, so that the installer will skip anaconda installation and will automatically install all the required dependencies through conda. If HOMER is already installed, please, USE `-c FLAG` to indicate the installation directory, and check the installation of the genomes. Otherwise, install manually the hg38 and mm10 genomes if they have not already been installed. Install the rest of the genomes which are not from human or mouse as well, if they are going to be used. Look at [Installation of other genomes](#) and [Installing HOMER](#) to see how to install a genome with HOMER.

NaviSE Genomes

NaviSE is programmed to allow the prediction of superenhancers based on any sort of genome, which is explained later.

Total implementation of NaviSE (functional chromosomal plots, Enrichr, GOEA, etc.) is only applied to hg38/19 in human and mm10/9 in mouse. The organisms in the following table include information about TF from HOMER and StringDB:

Name	Version	Scientific name	Common name
dm	3, 6	<i>Drosophila melanogaster</i>	Fruitfly
rn	4, 5	<i>Rattus norvegicus</i>	Rat
hg	18,19,38	<i>Homo sapiens</i>	Human
susScr	3	<i>Sus scrofa</i>	Pig
sacCer	2, 3	<i>Saccharomyces cerevisiae</i>	Yeast
rheMac	2, 3	<i>Macaca mulatta</i>	Rhesus macaque
tair	10	<i>Arabidopsis thaliana</i>	Arabidopsis thaliana
mm	8,9,10	<i>Mus musculus</i>	Mouse
galGal	4	<i>Gallus gallus</i>	Chicken
ce	6, 10	<i>Caenorhabditis elegans</i>	Caenorhabditis elegans
xenTro	2,3	<i>Xenopus tropicalis</i>	Xenopus tropicalis
danRer	7, 10	<i>Danio rerio</i>	Zebrafish
ci	2	<i>Ciona intestinalis</i>	Sea squirt
canFam	3	<i>Canis familiaris</i>	Dog
gorGor	3	<i>Gorilla gorilla</i>	Gorilla
panTro	4	<i>Pan troglodytes</i>	Chimpanzee

Finally, the use of any other genome will include the prediction and annotation of superenhancers, chromosomal plots, and statistical graphs. Apart from hg19/38 and mm10/9, any other genome must be installed in the system of files of NaviSE.

Installation of other genomes

In order to install other genomes, these are the steps to follow:

- If the genome appears in the table, in order to obtain information about HOMER TF, the genome must be installed in HOMER binaries. In order to install the genome, take the homer installation file `configureHomer.pl` and load it into the console. Then, install the genome by typing:
`perl PATH/T0/ConfigureHomer.pl -install XXX` where XXX is the genome. This step is also explained in the section [Installing HOMER](#).
- Downloading fasta assemblies and chrom sizes. Fasta assemblies are files with information of the genome of the organism. They are required to align the reads with Bowtie2 or other aligners. Chrom sizes is a file that contains information about the size of the chromosomes, which is used to create the chromosomal plot.
In order to download these files: (1) Head to <http://hgdownload.soe.ucsc.edu/downloads.html> and click on the organism. (2) Choose the desired genome version. (3) It will redirect to an ftp where several files are located. Choose the file ending with `.chrom.sizes` and one which end in `.fa.gz`. This last file should also contain the name of the genome or something similar. Put the chrom sizes file into the `Chrom_sizes` directory at NaviSE files. The fasta assembly file is compressed (into the `.gz` file) so it must be extracted with a compressor like `gzip` or `7zip`. The final file, `XXX.fa` must be renamed to `.fasta` and must contain the name of the genome. These steps are depicted in the figure [8](#)
- Downloading gene files. Gene files are files which provide with information about genes and are required by NaviSE to annotate the superenhancers. In order to download the gen files, head to Biomart web page (1), <http://www.ensembl.org/biomart/martview/> and choose the organism from the database. Then, establish some filters for genes, like genes with Entrez IDs (2). Then, select the attributes to be shown (3). Among the number of attributes, those that are ticked must be chosen. Finally, download the CSV file (4). The file should be `mart_export.txt`. Rename the file with the genome (`XXX.txt`) and place in in the `Genes` dir at NaviSE files. These steps are depicted in the figure [9](#)

The image shows a sequence of browser screenshots illustrating the download process for Sloth genome data from UCSC. The top screenshot shows the main UCSC Genome Bioinformatics page with a navigation menu and a 'Sequence and Annotation Downloads' section. A red box highlights the 'Sloth' link in the 'VERTEBRATES - Complete annotation sets' list. A red arrow points from this link to a second screenshot of the 'Sloth Genome' page, where another red box highlights the 'Full data set' link. A second red arrow points from this link to a third screenshot of a file index page, which lists various download files with columns for Name, Last modified, Size, and Description.

UCSC Genome Bioinformatics
 Home - Genomes - Blat - Tables - Gene Sorter - PCR - FAQ - Help

Sequence and Annotation Downloads

This page contains links to sequence and annotation data downloads for the genome assemblies featured in the UCSC Genome Browser. For quick access to the most recent assembly of each genome, see the [current genomes](#) directory. There are also automated scripts that must always reference the most recent assembly.

To view the current descriptions and formats of the tables in the annotation database, use the "describe table schema" button in the [database](#) page (no longer maintained) also provides descriptions of selected tables in the database.

All tables in the Genome Browser are freely usable for any purpose except as indicated in the README.txt files in the download directory. To view the current descriptions and formats of the tables in the annotation database, use the "describe table schema" button in the [database](#) page (no longer maintained) also provides descriptions of selected tables in the database. These data were contributed by many researchers. Please acknowledge the contributor(s) of the data you use.

VERTEBRATES - Complete annotation sets

- [Human](#)
- [Alpaca](#)
- [American alligator](#)
- [Armadillo](#)
- [Atlantic cod](#)
- [Baboon](#)
- [Bonobo](#)
- [Brown kiwi](#)
- [Budgerigar](#)
- [Bushbaby](#)
- [Green Monkey](#)
- [Guinea pig](#)
- [Hedgehog](#)
- [Horse](#)
- [Kangaroo rat](#)
- [Lamprey](#)
- [Lizard](#)
- [Malayan flying lemur](#)
- [Manatee](#)
- [Marmoset](#)
- [Platypus](#)
- [Rabbit](#)
- [Rat](#)
- [Rhesus](#)
- [Rock hyrax](#)
- [Sheep](#)
- [Shrew](#)
- [Sloth](#)
- [Squirrel](#)
- [Squirrel monkey](#)

Sloth Genome
 July 2008 (Broad/choHof1)

- [Full data set](#)
- [Annotation database](#)
- [LiftOver files](#)
- [Pairwise Alignments](#)
 - [Sloth/Mouse \(mm10\)](#)

Index of /goldenPath/choHof1/bigZips/

- Users are free to use the data in scientific papers analyzing particular genes and regions if the provider of these data (The Broad Institute) is properly acknowledged.
- The center producing the data reserves the right to publish the initial large-scale analyses of the data set, including large-scale identification of regions of evolutionary conservation and large-scale genomic assembly. Large-scale refers to regions with size on the order of a chromosome (that is, 30 Mb or more).
- Any redistribution of the data should carry this notice. 1. The data may be freely downloaded, used in analyses, and repackaged in databases.

GenBank Data Usage

The GenBank database is designed to provide and encourage access within the scientific community to the most up to date and comprehensive DNA sequence information. Therefore, NCBI places no restrictions on the use or distribution of the GenBank data. However, some submitters may claim patent, copyright, or other intellectual property rights in all or a portion of the data they have submitted. NCBI is not in a position to assess the validity of such claims, and therefore cannot provide comment or unrestricted permission concerning the use, copying, or distribution of the information contained in GenBank.

Name	Last modified	Size	Description
Parent Directory		-	
choHof1.2bit	16-Oct-2008 13:35	627M	
choHof1.agp.gz	13-Jul-2012 10:13	15M	
choHof1.chrom.sizes	07-Oct-2008 16:37	9.5M	
choHof1.fa.gz	13-Jul-2012 10:25	664M	
choHof1.fa.masked.gz	13-Jul-2012 10:34	435M	
choHof1.fa.out.gz	13-Jul-2012 10:15	89M	
choHof1.trf.bed.gz	13-Jul-2012 10:15	2.4M	
md5sum.txt	13-Jul-2012 10:40	304	
mrna.fa.gz	22-Feb-2017 23:19	2.3K	
mrna.fa.gz.md5	22-Feb-2017 23:19	45	
xenoMrna.fa.gz	22-Feb-2017 23:32	6.0G	
xenoMrna.fa.gz.md5	22-Feb-2017 23:32	49	
xenoRefMrna.fa.gz	22-Feb-2017 23:33	295M	
xenoRefMrna.fa.gz.md5	22-Feb-2017 23:33	52	

Figure 8: Download of files from UCSC

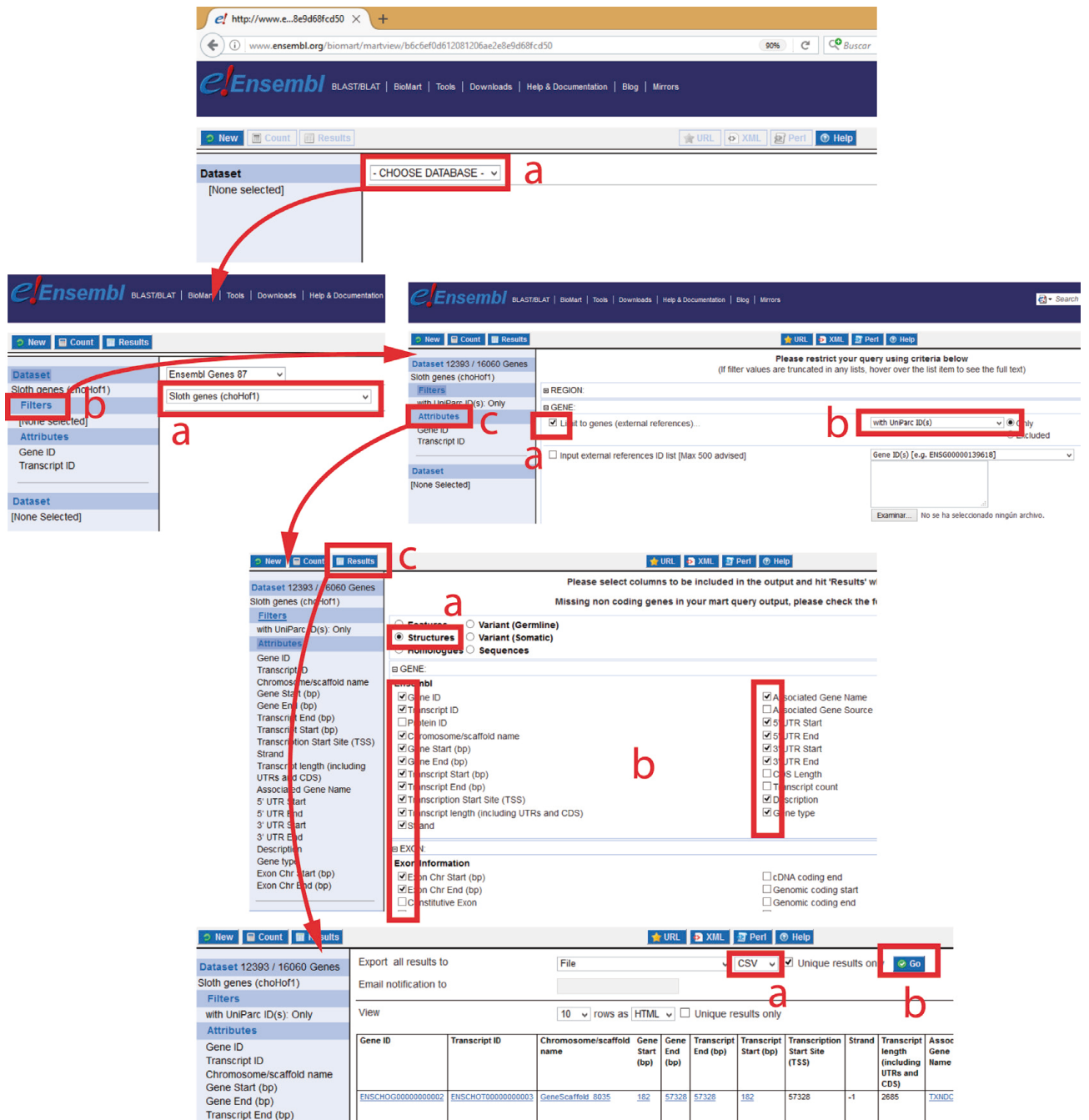


Figure 9: Download of files from Biomart

Commands

In order to run NaviSE, the python file `NaviSE.py` must be run in the terminal, adding all the required commands afterward. NaviSE requires python3.5, so the python file must be run with python. Hence, an example of a run of NaviSE should be like that:

```
python3.5 XXX/NaviSE.py [OPTIONS]
```

Where XXX is the path where `NaviSE.py` file is located. If you have automatically installed NaviSE or you have set the PATH to NaviSE, you only have to write:

```
python3.5 NaviSE.py [OPTIONS]
```

The commands allowed in user input are described below. The color scheme is the following:

(O/R) [-s][**-long-format**] (default-value) [input] : [Description]

(O/R) indicates if the argument is Required or is Optional. If a required argument is not inserted, it will throw an error. If an optional argument is not introduced, it will use the default value. [-s][**-long-format**] is the format of the variable to which you can assign your value. Remember that for [-s] a single hyphen at the start is required whereas for [**-long-format**] a double hyphen is required. (default-value) indicates which is the default value in case an optional variable is not declared. Lastly, [input] refers to the user input that will be assigned to that variable (sometimes an example of a value appears).

As for inputting, NaviSE will follow some recognition steps, so take this points into account so as not to get any error:

- If a path to a directory/file is introduced, the paths of the files must not contain any spaces. For instance, `/path to file/file a.png` should be corrected to `/path_to_file/file_a.png`. Also, it is recommended not to include symbols like (, ; ! ? % &), etc.
- In some cases, more than one option can be selected (like a gene list or several GSEA options). In that cases, each element must be separated by a **SPACE**, no commas, colons or semicolons. For instance, `[a b c d]` will be split to elements `['a', 'b', 'c', 'd']`, whereas `[a, b, c, d]` will be split to `['a,', 'b,', 'c,', 'd']`, or `[a,b,c,d]` will not be split because there are not spaces.

Warning: some of the programs referred in the commands may sound unknown. We describe them in detail in the [Running NaviSE](#) section.

Allowed commands are the following:

- (O) [-r][**-root**] (location of NaviSE.py) [XXX/NaviSE.py] : Directory where all NaviSE python files and other related files (logos for html, bowtie indexes, band files for chromosomal plots, etc. will be located.)
- (O) [-a][**-program-root**] (location of NaviSE.py) [XXX/Programs/] : Directory where ROSE, IGV and Gene Ontology directories are located. All directories must be located in the same directory.)

- (O) [-ch][**-conda-homer**] (location of NaviSE.py or home directory) [XXX/Programs/] : Directory where conda and homer files are located. By default it will search at /home directory as well as the directory where NaviSE.py is located, although it is recommended to include the exact location of anaconda and HOMER directories if there are many files at /home, since the search may take a while. There are several combinations of input:
 - If no directory is introduced, NaviSE will perform an automatic search. If it does not find anything, because files are not located at /home or NaviSE.py directory, or because conda might not be installed, please, locate the directories and include them at the command line. In order to search for conda or HOMER, write at the terminal, respectively, `which conda` and `which annotatePeaks.pl`. It will return something like `../anaconda3/bin/conda` and `../HOMER/bin/annotatePeaks.pl`. Therefore, you should add to the command line the directories `../anaconda/bin/` and `../HOMER/bin/`, or `../anaconda/` and `../HOMER/`.
 - If homer and conda share the same directory, you can write the common directory, although we recommend to introduce both specific directories to save time.
 - If both directories are introduced, NaviSE will allocate the **first directory to conda** and the **second directory to HOMER**, not the other way round.
 - If HOMER is not installed and you do not know the directory, it is possible to write `None`, and NaviSE will look for conda at /home or at the location of NaviSE.py. However, we recommend writing the directory to conda to save time.
 - If conda directory is to be introduced but HOMER is not installed (because it is not going to be run), the way to introduce the command is `XXX None` where XXX is the directory to conda. In this way, NaviSE recognizes that, with `None`, HOMER is not installed, and will not spend time looking for it.
- (R) [-i][**-input**] [XXX/DIR_OF_CHIPSEQ_FILES/]: Directory where all the ChIP-seq files will be located. Whether one file or multiple files are analyzed at once, this directory must be created, as all the result files will be created in that directory.)
- (O) [-o][**-output**] (/root-of-bams/SUPERENHANCERS/cell-mark/) [XXX/NaviSE.py] : Directory where Superenhancer results will be located. If it is blank it will be automatically completed, if it is a name (does not contain /) a directory will be created in the input directory (directory where bam files are located), and if the name is a route, it will create the route and will set the files there. In order to create subdirectories in the input directory, write `"../DIR1/DIR2"`.
- (R) [-n][**-file-name**] [H3K27AC]: Corresponds to the name of the file. Our recommendation is to put the name of the transcription factor or histone mark that is being analyzed. If there is other information to add, like cell types or other *index* names, we recommend using the command [-m] to assign that parameter. If more than one replicate is analyzed, like [H3K27AC_1, H3K27AC_2, H3K27AC_3], write the common name of them (H3K27AC), and NaviSE will take charge of determining the replicates by itself. NaviSE also allows the combination of different marks/samples with logical operators. The combinations allowed are the following:
 - AND: it takes the region of superenhancers which appears in both marks, and takes the minimum value from the intersection.
 - OR: it takes the junction of both samples, that is, the signal which appears in A or in B; and takes the maximal signal of both samples in case of the intersection.

- +: similarly to OR, it takes the junction of both samples, although it performs the sum of the signal instead of the maximal signal of the intersection.
- NOT: it takes the peaks which appear only in the first signal.
- -: similar to NOT, but although it subtracts the second signal from the first one. IF any pileup value is negative, it is converted to zero.
- XOR: it takes the signal that appears only in A or in B, but not in the intersection.
- SYM: similar to XOR, although the signal at intersection points may appear, as it performs the subtraction of the signal, not the logical negation.

A scheme of the logical operators is depicted in Fig. 10.

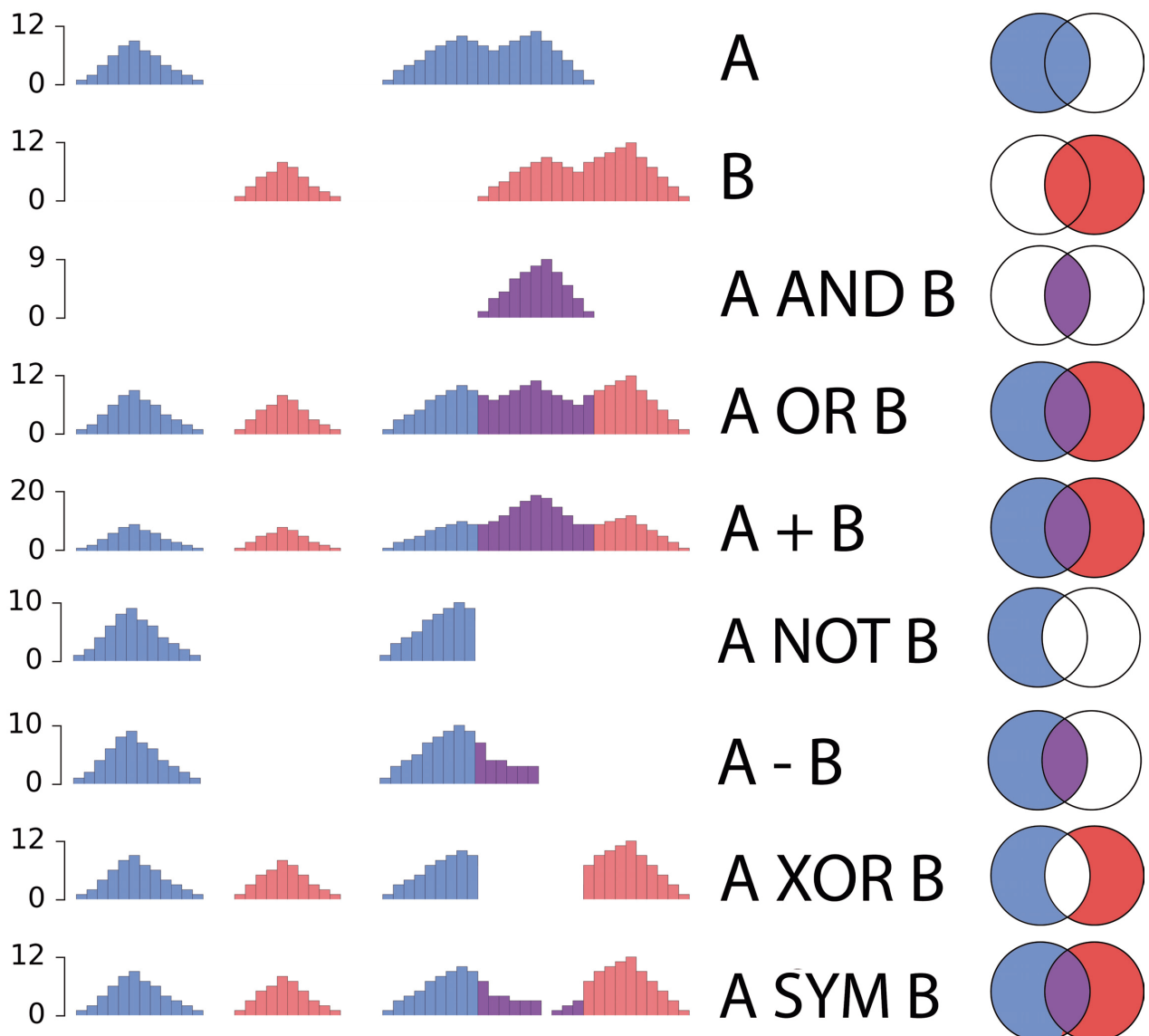


Figure 10: Logical operators

Logical operators work in pairs and sequentially, that is, if we write **A OR B NOT C**, NaviSE will first perform the junction between A and B, and then will remove the peaks from C.

- (O) [-m][**-cell-name**] (Nothing) [CD4 (A cell type, for instance)]: A secondary specifier for the filename.

Combining both [-n] and [-m] parameters, NaviSE will recognize all the files, and will name all the subsequent files with the format 'M_N' (for example, CD4_H3K27AC). This name structure will be used throughout the process and will appear as such in the report. If no [-m] is assigned, the final name will only be the correspondent to [-n].

- (O) [-c][**-control**] (Nothing) [CD4_CTRL]: The name of the control file(s). If more than one replicate exists, use the root name of all replicates, like with [-n]. In this case, there is no [-m] secondary specifier. If any, it has to be manually separated from the main name by an underscore.
- (R) [-g][**-genome**] [hg38]: Input the genome of the organism. Supported genomes: hg38, hg19, hg18 (might cause some problems) for human; mm10, mm9 for mouse.)
- (O) [-s][**-stitching-distance**] (12500) [12500]: The stitching distance ROSE will use for the stitching of MACS peaks. These stitched regions will be used later on to calculate the number of reads and will be ranked, resulting in the list of superenhancers.
- (O) [-d][**-tss-distance**] (2500) [2500]: The distance from the center of the peaks to the TSS ROSE will filter out of the analysis.
- (O) [-x][**-macs-threshold**] (MAX) [PER25]: The threshold for peak selection after peaks have been identified by MACS. There are several options:
 - [MAX]: NaviSE calculates the maximum bin from the histogram of the distribution of values (the kind of value is determined with the [-y] option). The number of optimal bins in order to select the maximum bin (which will establish the threshold) is selected according to the Freedman-Diaconis rule.

$$h = 2 \frac{IQR}{n^{1/3}}$$

The bin width is proportional to the interquartile range (IQR) and inversely proportional to the cube root of the size. Can be too conservative for small datasets, but is quite good for large datasets. The IQR is very robust to outliers.

- [PERXX]: XX is a value between 1 and 99. NaviSE calculates this value according to the percentile of values; that is, if [PER99], NaviSE will select the highest 1% of the peaks.
- [###]: ### is a number above zero. NaviSE will select the values above the threshold.
- (O) [-y][**-macs-choice**] (1) [3]: The statistic value upon which NaviSE will apply the threshold. Only accepts numerical values:
 - 0 -log10(p-val)
 - 1 -log10(q-val) (recommended)
 - 2 Pileup: Pileup height at peak summit.
 - 3 Fold enrichment: Fold enrichment for the peak summit against random/control background.
- (O) [-t][**-time**] (T) [F]: Makes a tabular (csv) report of how long has each process of the program taken.



Figure 11: List of colors for coloring.

- (O) [-P][*-processors*] (*auto*) [4]: The number of processors NaviSE will use for process multiprocessing. When [*auto*], NaviSE calculates the optimal number of processors according to how much memory each process consumes, using a maximum of the 80%.
- (O) [-M][*-mode*] (1) [0]: Mode in which NaviSE will run its commands.
 - 0 : NaviSE will run the basic commands (obtain superenhancer files, basic statistical graphs and chromosome plots, a few of the superenhancer snapshots and the html report).
 - 1 : NaviSE will run everything.
 - 2 : NaviSE will take some snaps and will do all graphs, but will exclude Enrichr, StringDB, HOMER and GSEA.'

Note: modes [0] and [2] are for computers that do not contain almost any processing capacity. However, if the computer has more than 4 or 6 processors, we recommend using the [1] mode, since even with little processors, the amount of time saved will be considerable for the amount of information this mode provides.

- (O) [-N][*-samples*] (30) [47]: The number of elements (e.g. number of snaps in mode 2, number of bars in the graph of GOEA, etc.) NaviSE will represent in the report.
- (O) [-C][*-colors*] (*Blue Red Green*) [*Pink Orange*]: List of color pairs that will be used for graph making. If less than three colors are chosen, NaviSE will choose the remaining at random. The list of colors is the following (Fig. 11):
- (O) [-D][*-dpi*] (450) [790]: The number of dots per inch of the graphs. We recommend a value around 300 or 400. A value higher than 700/800 or below 150 dpi is not recommended.
- (O) [-G][*-gsea-cutoff*] (*All*) [*PER75 SE 390 All*]: The choice (or list of choices) of superenhancers + typical enhancers that NaviSE will use for GSEA.
 - [*All/None*]: GSEA will be run with all superenhancers and typical enhancers.

- [PERXX]:XX is a value between 1 and 99. NaviSE calculates this value according to the percentile of values; that is, if [PER99], NaviSE will select the highest 1% of the peaks; and will run GSEA with the superenhancers + typical enhancers within that range.
- [##]: ## is a number above zero. NaviSE will select the lines corresponding to superenhancers or typical enhancers threshold.
- [SE/ONLYSE]: GSEA will be run only with superenhancers.

Note: We know that the number of superenhancers or typical enhancers varies a lot between samples and we don't know this beforehand, so using a number *per se* or the [PERXX] choice might be risky. Still, it is interesting to guess which this value might be (after all, if it is wrong NaviSE will pop this value out of the list), so you can input several values and select the results that best fit to your analysis.

On the other hand, it is not recommendable to use the [SE] option, as most of the times no signatures of GSEA are matched to this sample.

Taking these recommendations into account, the best choice is to make a range of thresholds, trying to exclude the highest number of typical enhancers as possible, but without reducing the number of matches too much.

- (O) [-S][–signatures] (All) [h c1 c3 c6]: The gene sets corresponding to signatures of MSigDB. Currently, no custom gene sets can be added. Options: "All" (all the signatures), "h" and "cX" being X from 1 to 7. These are the gene set category that each signature includes (according to the [MsigDB page](#)):
 - h hallmark gene sets: are coherently expressed signatures derived by aggregating many MSigDB gene sets to represent well-defined biological states or processes.
 - c1 positional gene sets: for each human chromosome and cytogenetic band.
 - c2 curated gene sets: from online pathway databases, publications in PubMed, and knowledge of domain experts.
 - c3 motif gene sets: based on conserved cis-regulatory motifs from a comparative analysis of the human, mouse, rat, and dog genomes.
 - c4 computational gene sets: defined by mining large collections of cancer-oriented microarray data.
 - c5 GO gene sets: consist of genes annotated by the same GO terms.
 - c6 oncogenic signatures: defined directly from microarray gene expression data from cancer gene perturbations.
 - c7 immunologic signatures: defined directly from microarray gene expression data from immunologic studies.

Warning: The signatures from MsigDB correspond only to human genes. Still, trying GSEA with mouse samples might lead to interesting results as well.

- (O) [-L][–gene-list] (Nothing) [SOX2 POU5F1 ACTN1 FN1 PI3K]: A list of genes determined by the user (may refer to genes they are interested in). NaviSE will recognize those genes and will mark them in bold in the chromosomal plots or in tables if they appear.
- (O) [-A][–aligner] (BOWTIE) [MOSAİK]: Choice of the aligner for the program to align the fastq files to sam files.
- (O) [-Z][–with-subpeaks] (True) [True]: Add the subpeak locations at the Genome Viewer graph. This feature is explained later.

Running NaviSE

For those users who are interested in knowing the details and the process NaviSE goes through, here is a detailed explanation of each process:

- a) **Sra to bam:** At this very first step, NaviSE recognizes files that contain the introduced filename and determines their format, as well as control files. Allowed **dataformats** are `.sra`, `.fastq`, `.sam`, `.bam` and `.bed`. After determining the formats, NaviSE will transform a superior format (`.sra`, `.fastq` or `.sam`) into `.bam`. If there are more than one formats, NaviSE will make a decision upon the number of files of each type. For instance, if the number of superior files is higher than the inferior ones, NaviSE interprets this as if the transformation was not complete, so it transforms all the superior files again.

Alignments are performed by bowtie2. Bowtie2 needs to create some index files for alignment of reads with the genome, those index files will be located at NaviSE directory. The first time NaviSE is run, bowtie2 will create these files, which may take around 3 hours. However, this is a once-in-a-lifetime process, since at the subsequent runs NaviSE will detect these files and bowtie2 will not need to generate new ones. However, if this files are moved, renamed or deleted bowtie2 will need to remake those files (it may take less time since some files are stored internally), so take that in mind. Also, if another genome is used, bowtie2 will create other files for that genome version.

If other aligners are used, the procedure is the same: if it is the first time that the aligner is used for one genome, the NaviSE will call the aligner to generate the index files and then it will align the reads using those index files.

Warning: Despite NaviSE transforming successfully `.bed` format to `.bam`, we recommend using a format that has not been aligned, such as `.sra` or `.fastq`, in order to make sure the processed files correspond to the correct version of the genome.

- b) **FastQC:** FastQC is a program that performs a quality analysis of `.fastq` files (Fig. 12). Then, it creates a report in which several quality parameters are included, such as per base quality, GC content, or presence of adapters.
- c) **Combination of bams:** If there is more than one replicate or control, NaviSE will combine all the `.bam` files into one, and use this file for the analysis. If subsequent analysis are performed with that file and NaviSE detects this combined file, NaviSE will not combine the files again. Therefore, if any modification is performed to any of the original files, mind deleting the combined file as well.
- d) **MACS:** MACS is a software that calculates peaks from bam files. Those peaks indicate the presence of a histone mark or the binding of the protein/transcription factor ChIP-seq analysis is performed with. If a control is introduced, MACS will use the information from control signal to calculate the peaks from the sample. Instead, if no control is introduced, MACS will use a precalculated background.

Once MACS has determined the peaks from the sample, peak files are processed for Superenhancer prediction by ROSE. In this processing, peak values below a threshold determined by $[-x]$ and $[-y]$ are excluded.

- e) **SE prediction:** NaviSE uses the algorithm developed by Young to predict the presence of superenhancers in a sample (Fig. 13). The algorithm is a stitching algorithm, that is, given a

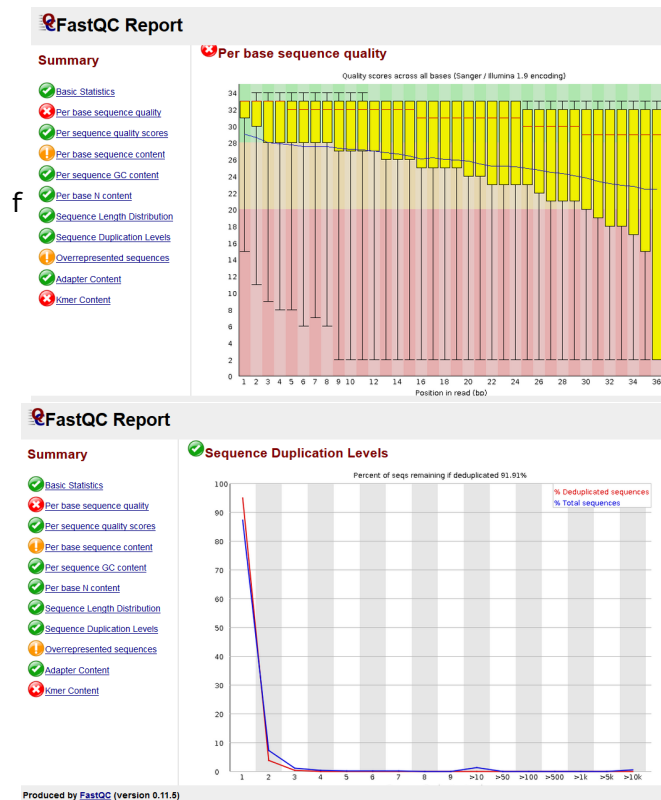


Figure 12: FastQC analysis

file that contains the location of enhancers (in our case, the processed peak file from MACS), NaviSE "stitches" those enhancers separated less than a threshold value. Out of these "stitched" enhancers, NaviSE ranks them by the number of reads that fall within that region. Finally, NaviSE establishes a cutoff, so the "stitched" enhancers falling within that range will be considered as Superenhancers. (Image from Sebastian Pott & Jason D Lieb. What are super-enhancers. Nature Genetics 47,812(2015). doi:10.1038/ng.3167)

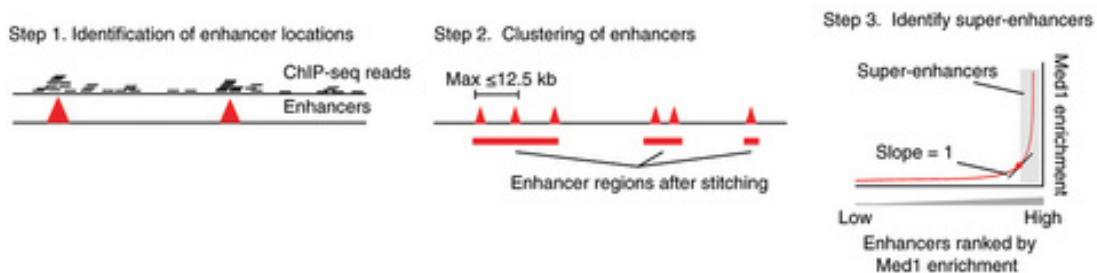


Figure 13: SE prediction algorithm.

f) **Gene annotation:** once the superenhancer locations are determined, each superenhancer is assigned a gene by proximity (independent of the strand of the gene). The following columns are included:

- Overlapping Genes: indicates genes that are overlapped with the superenhancer.
- Proximal Genes: genes that are next to the superenhancer but are not overlapped, and whose TSS are found at less than 250 kb from the superenhancer.

- Closest Gene: the closest gene to the superenhancer.
- Gene Type: the category to the gene corresponds (ncRNA, protein-coding, snRNA, etc.)
- Gene Description: brief description of the closest gene.

g) **Subpeak annotation:** in this subsection, further columns are created which expand information about superenhancers by adding data corresponding to the MACS peaks within each superenhancer. This information is contained in several columns:

- Number of subpeaks: number of subpeaks that each superenhancer has.
- Loci and TSS locations: locations of the subpeaks and the distance from each subpeak to the TSS of the Closest Gene.
- SE Status, INS, OUTS, Percentage OUTS and Enhancer Type: SE Status, INS and OUTS relate to those subpeaks that fall within the range of the TSS-threshold determined by $[-d]$. From these values, a percentage of OUTS is calculated and, from this value a Enhancer Type is assigned, among 3 possibilities: *Mixed* (it contains OUT and IN subpeaks), *Pure* (it contains only OUT peaks) and *Only TSS* (it contains only IN peaks).

h) **Snaps:** NaviSE takes two snapshots of each superenhancer, one of them called *near* and the other one called *far*. Both options extend the locus begin and end following this equation:

$$x'_0, x'_f = x_0 - \frac{(x_f - x_0) \cdot (k - 1)}{2}, x_f + \frac{(x_f - x_0) \cdot (k - 1)}{2}, \quad x_f > x_0$$

where k is 1.2 for *near* and $\frac{400}{(x_f - x_0)^{0.34}}$ for *far*. If more than one sample is plotted, then an additional *Preview* snap is added, which contains the first sample; and which is added to the html final report (Fig. 14).

i) **HOMER motif finding:** HOMER (Hypergeometric Optimization of Motif EnRichment) is a suite of tools for Motif Discovery and NGS analysis. NaviSE uses HOMER in order to identify motifs of regulatory elements (mainly transcription factors) that are specifically enriched in the loci of superenhancers, relative to the loci of typical enhancers (which will be used as background). As a result, HOMER writes a list of motifs enriched in superenhancers and another list of *de novo* motifs, that is, a novel algorithm developed by HOMER which finds motifs for which their binding element is unknown, and tries to determine which is this element. Results of HOMER are explained in [HOMER analysis](#).

j) **Gene Ontology Enrichment Analysis (GOEA):** GOEA is an analysis performed over the set of superenhancer genes. A GOEA results in a list of Gene Ontologies, that is, sets of genes belonging to a certain metabolic/cellular pathway, to which the set of superenhancer genes is enriched in contrast to the background gene set. A result from a GOEA is further explained in [GOEA](#).

k) **Enrichr and StringDB results:** In this subsection, we extract data submitted to [Enrichr](#) and [StringDB](#) webpages. Enrichr comprises a number of databases, such as Human/Mouse Gene Atlas, ChEA/ENCODE consensus TF from ChIP-X, TRANSFAC and JASPAR PWMs, or Reactome/Wikipathways/KEGG pathways. StringDB is a protein-protein interaction (PPI) database which establishes PPI networks, based on literature-determined interactions or predicted interactions. In both cases, NaviSE submits the superenhancer set and collects and processes that information. Results from Enrichr in [Enrichr results](#) and from StringDB in [StringDB results](#). Finally, all the raw data from Enrichr is processed for an easier interpretation for the user, or filtered (selects organism-specific results).

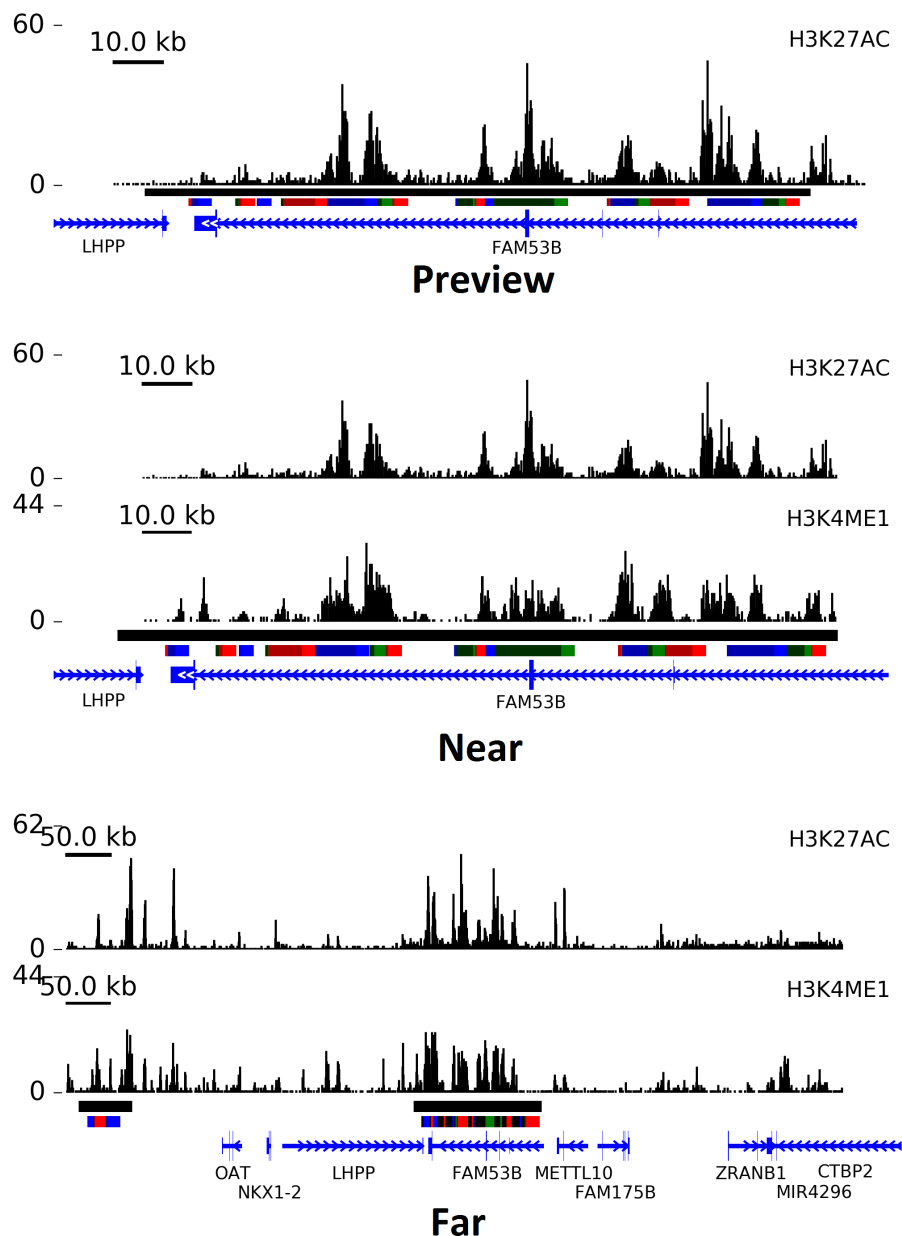


Figure 14: Genome Viewer sample at preview, near and far distances.

l) **Graphs and chromosomal plots:** The graphs production comprises a number representations about statistical values or properties shown before. These graphs will be explained in depth in [NaviSE Graphs](#). They also include barplots for easier interpretation of GSEA or Enrichr results (explained in their correspondent sections).

As for chromosomal plots, which are further explained in [Chromosomal plots](#), they are representations of superenhancer loci located in a karyotype. These plots allow users to make a first impression on how near or far two superenhancers are apart, or if a chromosome is depleted or enriched in superenhancers. In one NaviSE run three plots are generated:

- Simple plot: it just shows the loci location and the chromosome enrichment/depletion.
- Rank plot: loci are colored according to their rank. Several percentiles are represented,

so superenhancers falling within a percentile will be colored with its corresponding color.

- Closeness plot: this plot represents the range of proximity between superenhancers (< 5 Mb for instance) which will be colored with its corresponding color. This plot is incredibly useful to tell apart clusters of superenhancers, which may look like one superenhancer. For a number of superenhancers within a chromosome, $x_1, x_2, \dots, x_{a-1}, x_a, x_{a+1}, \dots, x_{f-1}, x_f$, their distance is determined by the following formula:

$$d = \begin{cases} x_2 - x_1 & \text{for } x_1 \\ \min(x_{a+1} - x_a, x_a - x_{a-1}) & \text{for } x_a \\ x_f - x_{f-1} & \text{for } x_f \end{cases}$$

In all the plots a p-value for an enrichment score is calculated, which determines whether a chromosome is enriched (λ for $p < 0.05$ and $\lambda\lambda$ for $p < 0.01$) or depleted (Υ for $p < 0.05$ and $\Upsilon\Upsilon$ for $p < 0.01$). This p-value is calculated by a binomial approximation of the hypergeometric distribution, where N is the number of genes in the whole genome, K is the number of superenhancers in all chromosomes, n is the number of genes in a chromosome and k is the number of superenhancers in a chromosome. This hypergeometric distribution is approximated to a binomial distribution ($h(k; K, n, N) \rightarrow b(k; K, p)$; $p = \frac{n}{N}$), so the p-value for depletion is the cumulative distribution function for this binomial approximation and the p-value for enrichment is survival function.

- m) **GSEA (Gene Set Enrichment Analysis):** GSEA is an analysis similar to GOEA, is a computational method that determines whether an a priori defined set of genes shows statistically significant, concordant differences between two biological states. In our case, the defined gene sets are the signatures from MsigDB (explained in [signatures](#)), and the two biological states are the superenhancers and the typical enhancers.

In our case, a similar approach is followed: both superenhancers and typical enhancers are ranked by their signal, the user filters out as many lines of these rank as it is established with the parameter [-G], and those genes that match the genes from a gene set from a signature are marked as positive. Then, positive matches are given a score according to their position in the list and a GSEA plot is drawn, which shows a curve that represents how "fitted" the ranking of genes is to the gene set. Further values representing the overall score of this "fitness" are also calculated.

GSEA results will be further analyzed in [GSEA results](#). For further information about gsea, we recommend reading the [following article](#).

- n) **Writing HTML report, deleting files and writing the timetable:** This is the last part of the analysis. NaviSE gathers all the information into a user-friendly html interface through which the user can navigate and access all the aforementioned information, and which is discussed in detail with an example in [NaviSE output](#). Obviously, all the information (tables, graphs,...) will be available in their respective files and directories in case the user wants to publish the figures or extract some data.

Finally, NaviSE removes some intermediate files which have no relevant information for the user; and creates a timetable (csv) in which the time taken for each part is shown.

Warning: Some of the processes that the timetable shows are not mentioned in this section, have a different name or belong to more than one section. However, the information is relevant and the user should have no problem in recognizing each process of the table.

If the timetable (which should appear in **FILES/timetable-DATE.csv**) is not present, it might be due to an error during processing. Also, if not all the subprocesses are present, this is due to an error or, simply, because that process was not run (because it was run before or because the selected mode restricts the process).

Parallelization of NaviSE

One of the main characteristics of NaviSE is its parallelization process, which considerably reduces the processing time. Currently, NaviSE parallelizes the most consuming processes, like e), f), g), h) and m), as well as minor processes such as a), c), d), in which the parallelization process is notorious at cases with multiple samples.

NaviSE determines the optimal number of processes, k , compatible with the computer resources. Such resources are the parallel processing capability of the computer measured as the number of cores, C , and the total main memory, M in GB. NaviSE optimizes automatically, for each processing task i , the number of processes, k_i :

$$k_i = \min(C, C_u, \lfloor M/m_i \rfloor, l_i) \quad (1)$$

where C_u is the maximum number of cores reserved by the user to run NaviSE, m_i is the memory, measured in gigabytes (GB), needed to run one process in task i , $\lfloor \cdot \rfloor$ is the floor operator and l_i is the cardinal of $D_i = \{d_1, d_2, \dots, d_m\}$ which is the set of *chunks* of distributed elements to be processed in task i . If $l_i > k_i$, the first k_i chunks are distributed to k_i cores. The distribution of information (SE peak distribution profiles, number of gene sets for GSEA, chromosomes for Superenhancer prediction) to be parallelised is based on a cyclic algorithm, implemented in Python. For the ordered set $S_i = \{s_1, s_2, \dots, s_n\}$ of information elements, the set $P_i = \{1, \dots, k_i\}$ of processes and for the set D_i (chromosomes, gene sets, positions on a list) to be distributed across processors, we define D_{pi} as the *chunk* of the task i that is assigned to each processor p :

$$D_{pi} = \{d_j \mid \forall d \in D_i, p \in P_i, j \in \{1, \dots, l_i\}, j \bmod k_i = p\} \quad (2)$$

where \bmod is the module operator. Once the *chunk* D_{pi} is constructed, the subset of information elements $S_{D_{pi}} \subset S_i$ will be defined depending on the type of process which is being parallelised.

The list of parallelised tasks is $i = \{\text{STIT}, \text{SNAP}, \text{GSEA}, \text{HOMER}\}$. In the case of SE prediction (STIT), the input table with peak coordinates from MACS (S_{STIT}) is divided in k_{STIT} files, calculated with Equation 1, with $m_{\text{STIT}} = 2$ GBs. Here, $D_{p,\text{STIT}}$ represents the groups of chromosomes that will be processed in each $p \in P$, and $S_{D_{p,\text{STIT}}}$ is the *chunk* of $s \in S_{\text{STIT}}$ elements which share the same chromosome from each group of chromosomes from D_p . In this case, $D_{\text{STIT}} = \{Y, 22, \dots, X, \dots, 2, 1\}$ (for human), i.e., the chromosomes are arranged in increasing length order, so that the distribution of $D_{p,\text{STIT}}$ is balanced across processors. For a better understanding of the process, an example is developed in Figure 15.

In the case of SE signal profile snapshot parallelisation, $S_{\text{SNAP}} \equiv D_{\text{SNAP}}$, is the set of SE *loci*. Hence $D_{p,\text{SNAP}}$ contains all the *loci* that fulfill Equation 2, based on k_{SNAP} with $m_{\text{SNAP}} = 2$ GBs.

In the case of GSEA parallelisation, S_{GSEA} is the set of genes ranked by SE score and D_{GSEA} is the set of combinations (GSEA signatures \times GSEA cutoffs). Therefore, $D_{p,\text{GSEA}}$ contains all the combinations that fulfill the Equation 2, based on k_{GSEA} with $m_{\text{GSEA}} = 2$ GBs.

The parallelisation of all these cases has been implemented with the *multiprocessing* module of Python. In the case of HOMER parallelisation, we took advantage of the HOMER parallelisation capabilities already implemented in HOMER, with the number of processes k_{HOMER} , optimized by Equation 1, with $m_{\text{HOMER}} = 2$ GBs.

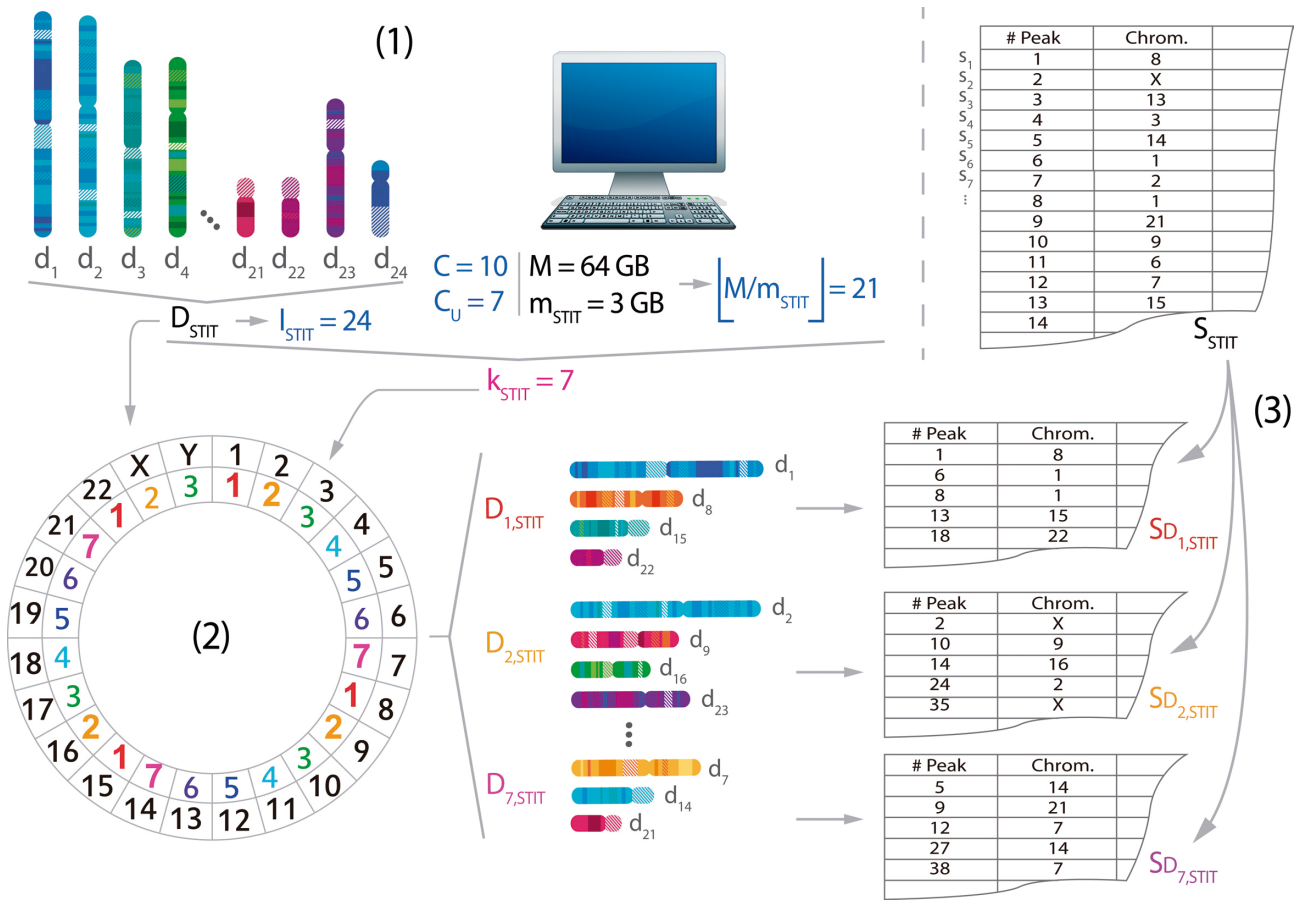


Figure 15: **Scheme of parallelisation of SE prediction.** (1) Determination of the number of processes (k_{STIT}) based on Equation 1, for which the number of cores (C) is 10, the maximum number of cores allocated (C_u) is 7, the memory of the computer (M) is 64 GB, the memory allocated to ROSE (m_{STIT}) is 3 GB and the cardinal (l_{STIT}) of the set of chromosomes ($D_{STIT} = \{d_1 = 1, d_2 = 2, \dots, d_{22} = 22, d_{23} = X, d_{24} = Y\}$) is 24. The calculated value is $k_{STIT} = C_u = 7$. (2) Construction of chunks based on Equation 2. Since $k_{STIT} = 7$, the set of chromosomes D_{STIT} is divided into 7 subsets or chunks: $D_{1,STIT} = \{d_1, d_8, d_{15}, d_{22}\}$; $D_{2,STIT} = \{d_2, d_9, d_{16}, d_{23}\}$; \dots ; $D_{6,STIT} = \{d_6, d_{13}, d_{20}\}$ and $D_{7,STIT} = \{d_7, d_{14}, d_{21}\}$. (3) Assignment of information elements. In the case of ROSE, assigned elements are MACS peaks (inferred as enhancers). After the assignment of the subsets $D_{1,STIT}$, $D_{2,STIT}$, etc., the set of MACS peaks, $S_{STIT} = \{s_1, s_2, \dots\}$ is divided into 7 subsets of elements, $S_{D_{1,STIT}} = \{s_1, s_6, s_8, \dots\}$, $S_{D_{2,STIT}} = \{s_2, s_{10}, s_{14}, \dots\}$, \dots , $S_{D_{7,STIT}} = \{s_5, s_9, s_{12}, \dots\}$. Finally, each subset of elements is simultaneously processed by ROSE, all the 7 subsets of stitched enhancers are combined into one file and the SE rank is performed.

NaviSE output

NaviSE outputs a huge amount of heterogeneous data, which is explained thoroughly in this section. We will use an example run of human embryonic stem cells with the H3K27Ac histone mark.

The following commands were introduced in the command prompt:

```
python3.5 /media/labcombio1/8TB-HD/Alex/Programs/NaviSE/NaviSE.py
-a /media/labcombio1/8TB-HD/Alex/Programs/ -i /media/labcombio1/8TB-HD/Alex/hESC/
```

```
-n H3K27AC -m hESC -c CTRL -g hg38 -s 12500 -d 2500 -x MAX -y 1
-t T -p auto -M 5 -N 35 -C 'Blue Red Green' -D 450
-G 'se per80 all' -S 'h c1 c2 c3 c4 c5'
-L 'POU5F1 OCT4 NANOG SOX2 KLF4 ESRRB BRD4 PRDM14 SMAD3 TCF3 ZMYND8 RNU2-1'
```

The main interface of NaviSE consists of a top navigation bar showing the different subcommands; and also a left navigation bar (sidebar) which contains the different subsections of each subcommand from the top navigation bar.

Main page

The main pages, which contains NaviSE logo on the top navigation bar, contains a small table indicating main characteristics, such as the sample name, cell name or number of superenhancers (Fig. 16). On the other hand, the sidebar includes all the chromosomal plots.

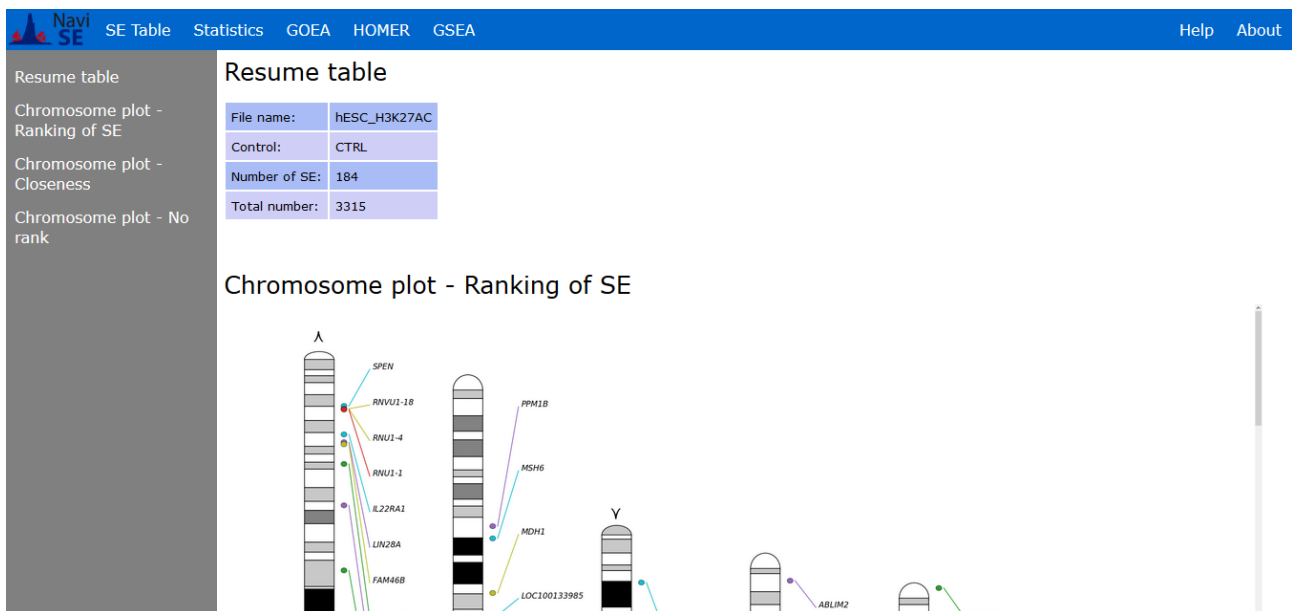


Figure 16: Main window.

The user can click on each name of the chromosomal plot, which will redirect to the correspondent superenhancer at the [SE table](#) section (Fig. 17).

Here is an example of some chromosomes from the chromosomal plot defined by closeness (Fig. 18):

SuperEnhancer table

SE table includes two subsections: *Complete table of SE*, which can be found at the **SUPERENHANCERS/ SUPER_CELL_NAME-CONTROL/FILES/Annotated_SE_table.csv** file and *Table of SE*, which can be found at the same file. The complete table contains extra information derived from the reduced table, and includes several columns, most of them previously explained at f) and g) paragraphs from [Running NaviSE](#).

The reduced table includes seven columns (Rank, Gene, Locus, SE Score, # Subpeaks, Snap and Zoom out.); shown in the Fig. 19

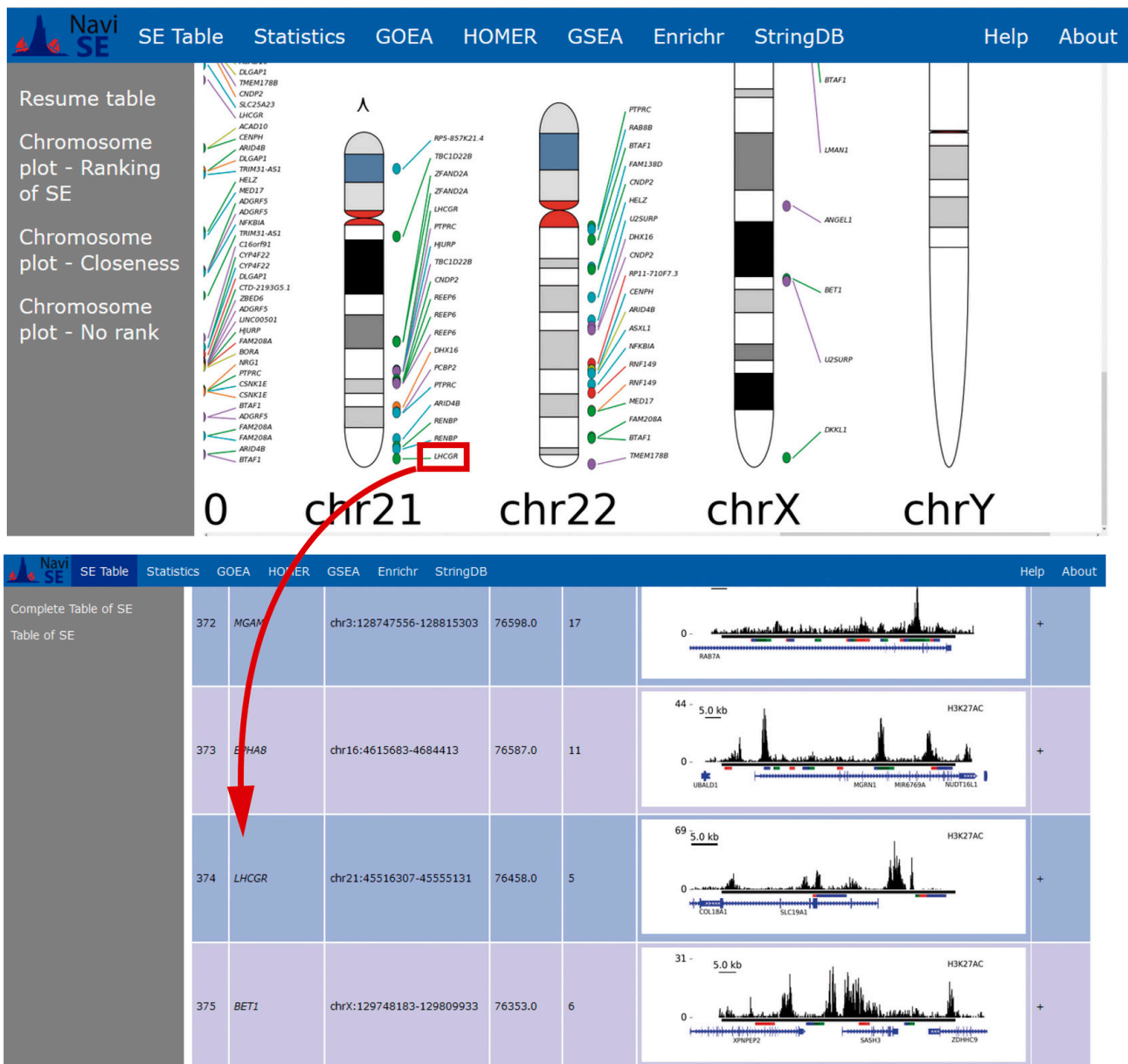


Figure 17: Main window 2.

Moreover, all those names which are included in the gene list determined by [-L] will appear in bold, for easier identification. Focusing on the two last columns, each superenhancer will contain a screenshot of the bam reads from that region. Clicking on the '+' symbol on the *Zoom out* column will redirect to the zoomed out screenshot of the superenhancer area.

Gene and *Locus* columns contain clickable links that will redirect to the [Genecards](#) website of that gene and to the [UCSC Genome Browser](#) website showing the locus region (Fig. 20).

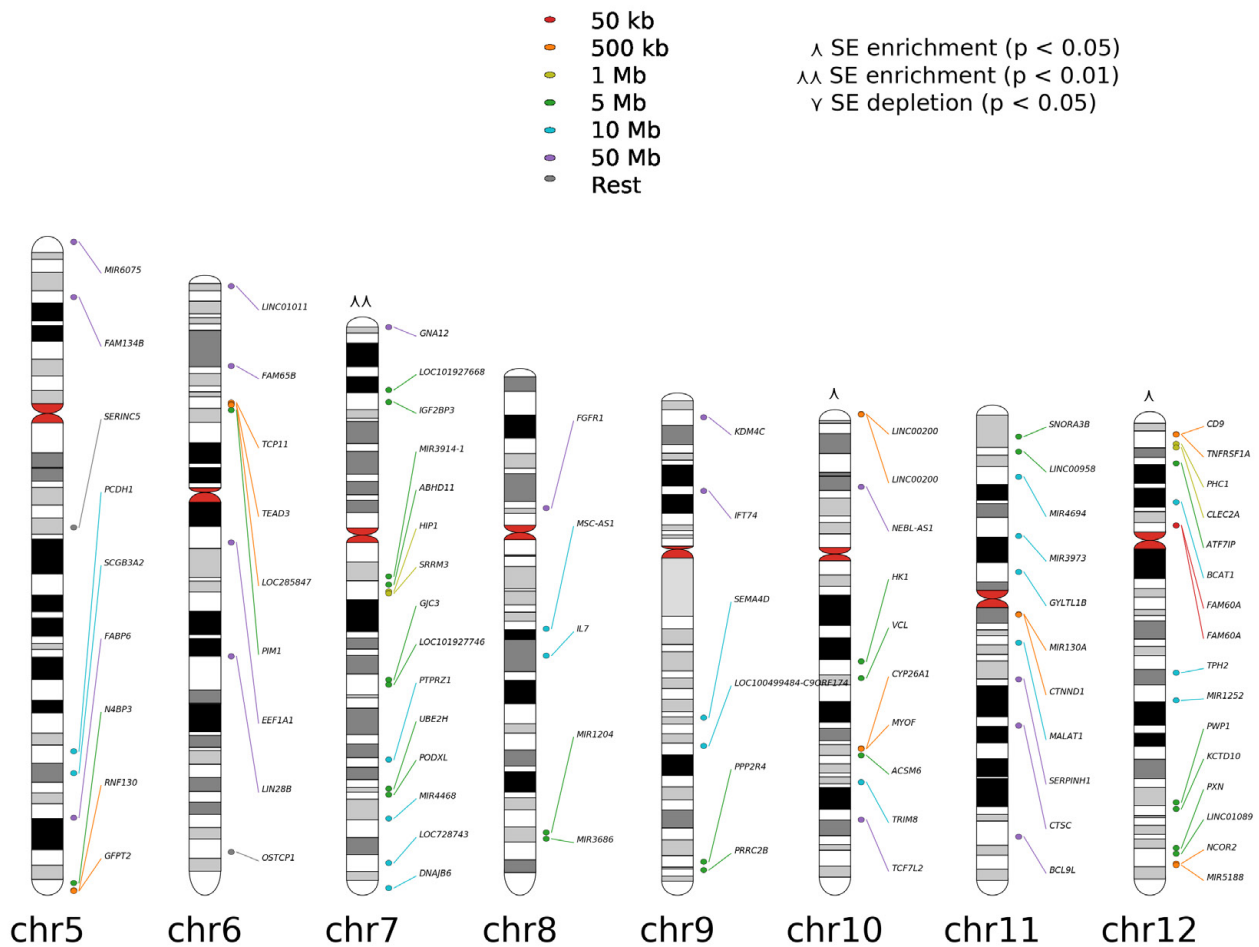


Figure 18: Chromosomal plot by closeness.

NaviSE Graphs

NaviSE implements a series of graphs which allow the user to obtain information related to the superenhancers in the sample. Those graphs are located in the *Statistics*. The sidebar contains all the accessible graphs, each of which is located in the **SUPERENHANCERS/SUPER_CELL_NAME-CONTROL/GRAPHS/** directory. If clicked on the graph, the image of the graph is displayed for easier observation (Fig. 21).

The included graphs are:

- Ranking by SE score:** This graph could be considered as the most representative graph of the distribution of superenhancers. The superenhancer score is represented against the rank of each superenhancer, which follows a *hockey stick* distribution (Fig. 22).

Superenhancers are painted in a darker color, while typical enhancers are painted in a lighter color. Generally, a *hockey stick* distribution in which the curve is more pronounced indicates that the resolution of the "technique/histone mark/DNA binding protein" is higher.

- INS and OUTS:** Ins and Outs graphs contains two subgraphs. The first one shows the percentage of superenhancers or typical enhancers that contain any of the types of regions (pure,

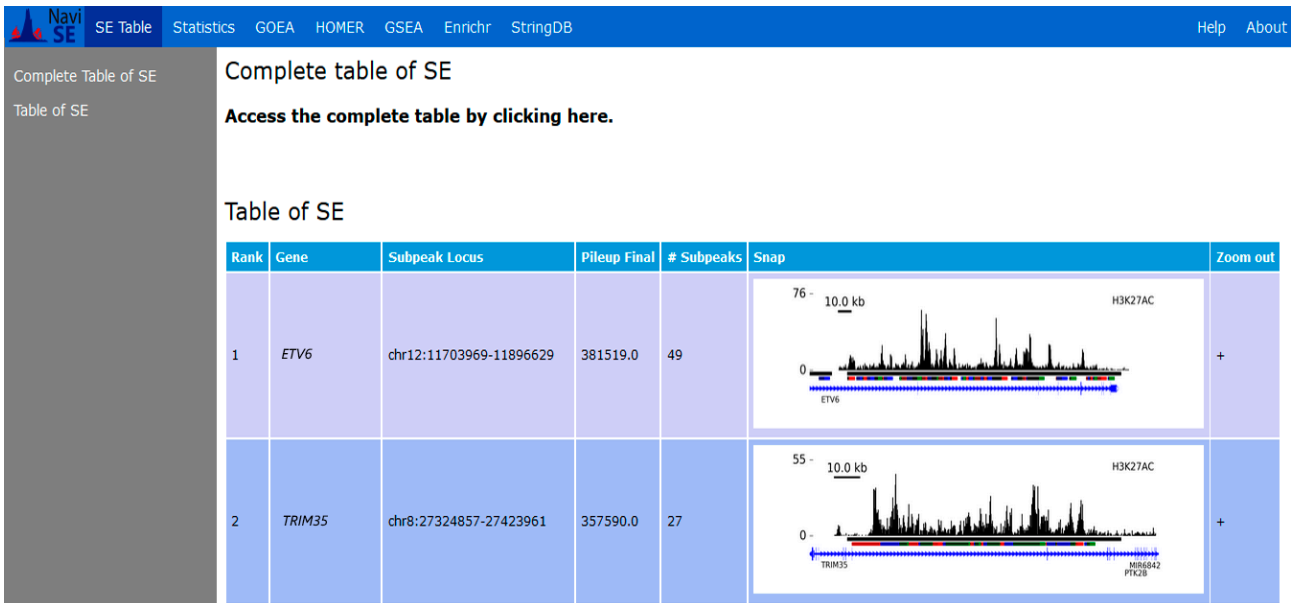


Figure 19: Superenhancer Table

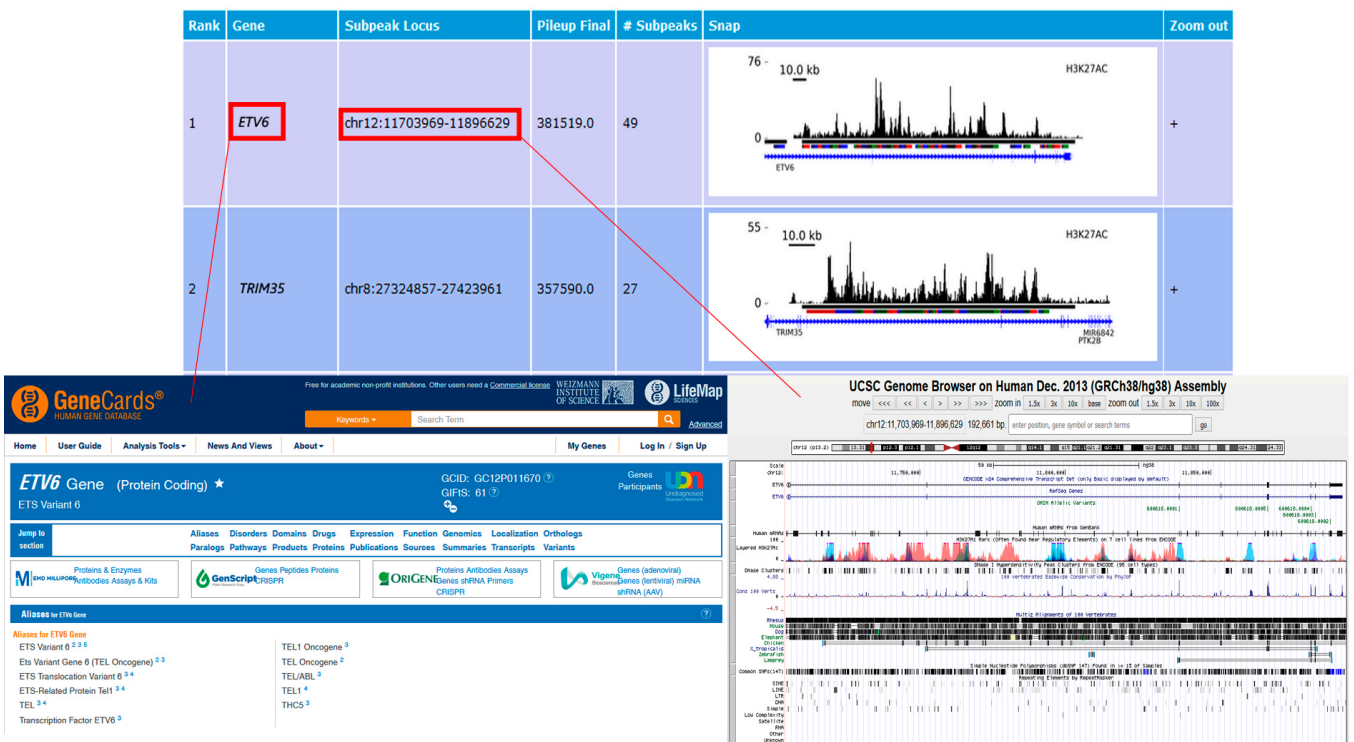


Figure 20: Superenhancer Table with links to GeneCards and UCSC Genome Browser

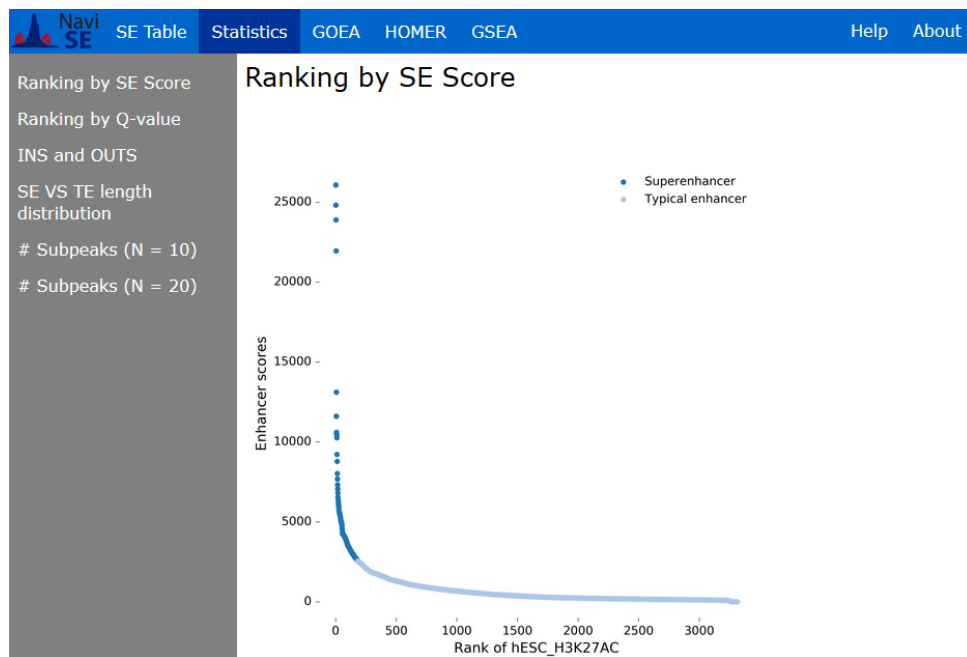


Figure 21: Window of SE Statistics.

mixed or only TSS). The second graph shows out of all types or out of mixed" types, which is the distribution of OUT subpeak in the regions. (Fig. 23)

In this example, we may observe that the number of pure superenhancers is diminished in comparison with typical enhancers, consistent with the fact that H3K27Ac is located in both TSS and enhancer regions. As for the amount of ins and outs in each superenhancer/typical enhancer, there are no statistically significant differences between both samples.

- **Length distribution:** This graph shows in a double histogram and a scatter plot the distribution of superenhancer and typical enhancer length and pileup (number of BAM reads) (Fig. 24). This graph is also developed for subpeaks from superenhancers and typical enhancers. The histogram lying on the X axis of the scatter corresponds to the length of superenhancers/typical enhancers; and the histogram on the Y-axis corresponds to the pileup.
- **Number of subpeaks:** This graph simply shows the distribution of the number of subpeaks superenhancers and typical enhancers have. (Fig. 25) Typically, typical enhancers show a [zipfian distribution](#) while superenhancers show a [chi-square](#)-like distribution, indicating that the number of subpeaks in superenhancers is clearly displaced in comparison with typical enhancers.

GOEA results

GOEA results includes all the results present at the **GENE_ONTOLOGY** directory. At first, the significant terms, that is the amount of highest-scoring positive GOEA terms (determined by the $[-N]$ parameter) will show in a barplot representing the rank VS $-\log(p\text{-value})$. On the other hand, a table with all the GO terms with a p-value smaller than 0.01 will be shown. GOEA discerns three different terms: biological process, molecular function and cellular component, each of which will appear as a .png file inside **GENE_ONTOLOGY** (Fig. 26).

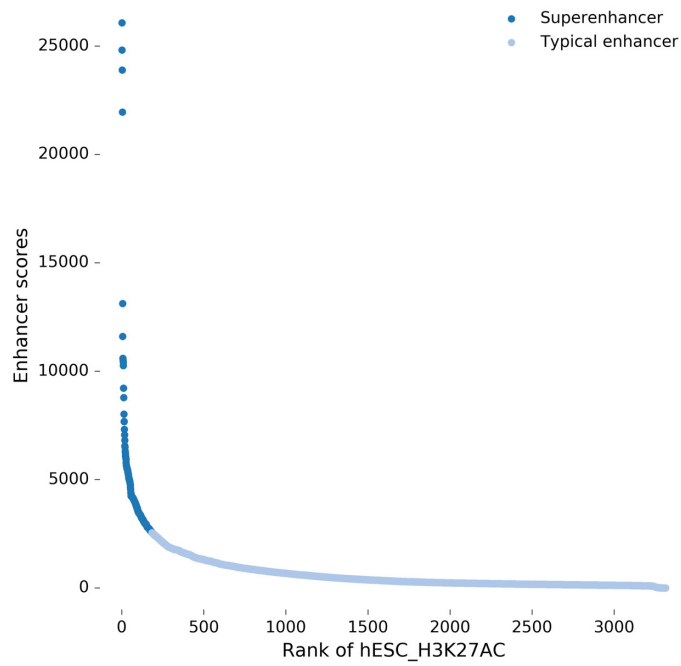


Figure 22: Rank of Superenhancers.

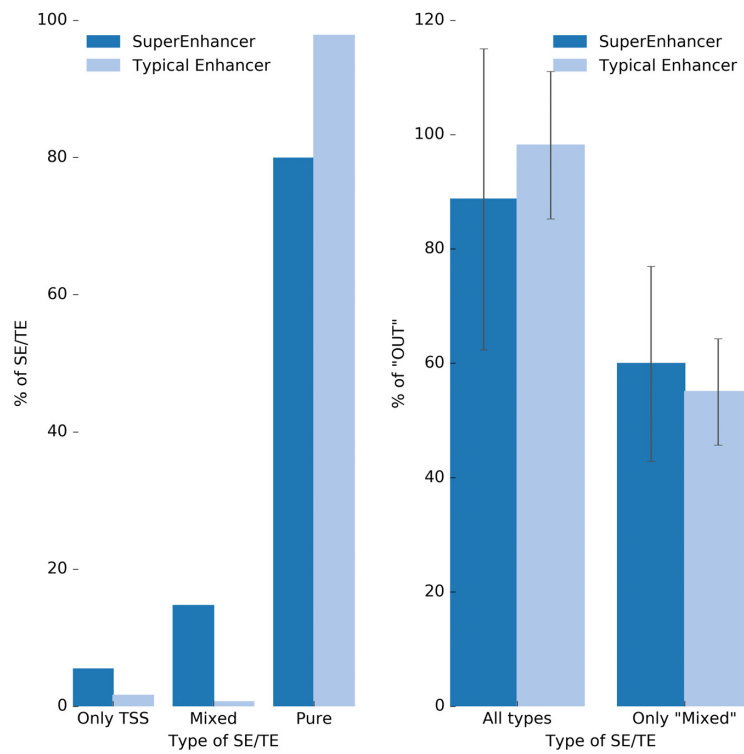


Figure 23: Graph of INS and OUTs regions.

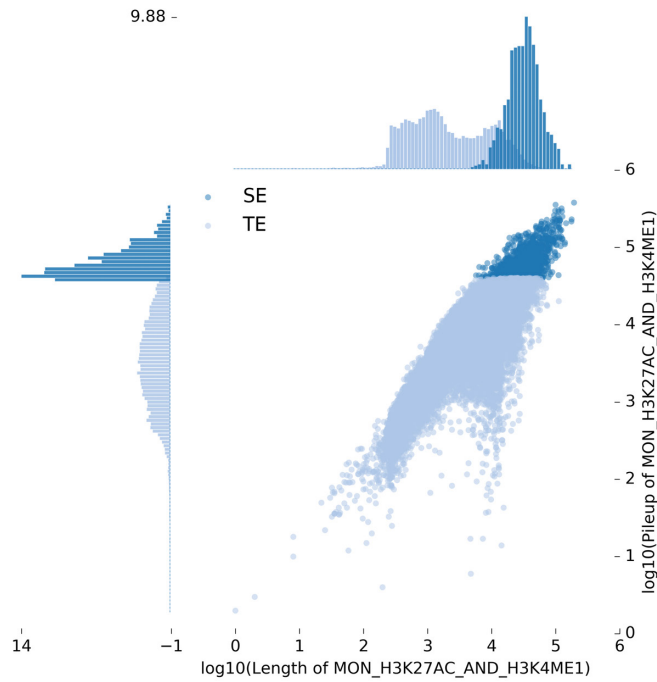


Figure 24: Graph of length/pileup distribution.

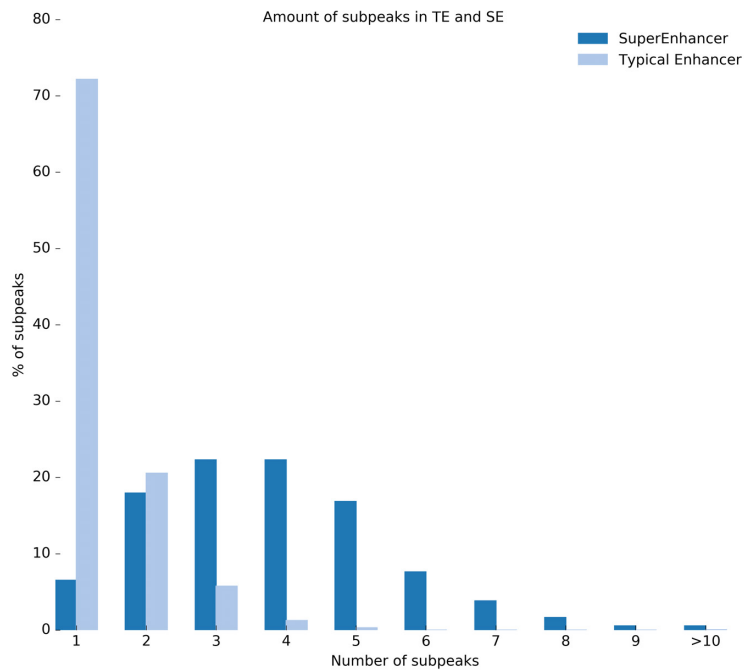


Figure 25: Graph of number of subpeaks.

These files represent a graph that links different GO terms with the positive GO terms in a way that, the higher the position in the graph, the more general the GO term is. Usually, these three graphs

Note: images from GO terms may contain genes that result interesting to you and want to access to some information, like GeneCards or *SE table* data, about them. If so, it is possible to search a superenhancer in *SE table* by going to that tab in the navigation bar and pressing **Ctrl + F**, which will prompt a search box.

HOMER analysis

The HOMER analysis section includes two tables, one for known motifs and another one for de novo motifs. The first one, HOMER known motifs contains the following distribution (Fig. 27):

- *Rank* of the motifs.
- *Motif* : LOGO of the motif
- *Name*: Name of the transcription factor / DNA binding protein that binds to that motif. It also includes a GSE number from Gene Expression Omnibus related to the experiment. Transcription factors are linked to their respective GeneCards page.
- *P-value* related to the enrichment of the sequences in superenhancers VS typical enhancers.
- *% of Target sequences with Motif* and *% of Background sequences with Motif*, being Target the superenhancers and the Background the typical enhancers.

Rank	Motif	Name	P-value	% of Targets Sequences with Motif	% of Background Sequences with Motif
1		Maz(Zf)/HepG2-Maz-ChIP-Seq(GSE31477)/Homer	1e-19	44.78%	28.33%
2		Nlx2.5(Homeobox)/HL1-Nlx2.5.biotin-ChIP-Seq(GSE21529)/Homer	1e-18	54.65%	37.94%
3		Klf4 (f)/mES-Klf4-ChIP-Seq(GSE11431)/Homer	1e-18	16.45%	6.68%
4		NF1-halfsite(CTF)/LNCaP-NF1-ChIP-Seq(Unpublished)/Homer	1e-18	49.93%	33.75%
5		EBF1(EBF)/Near-E2A-ChIP-Seq(GSE21512)/Homer	1e-18	40.06%	24.91%
6		Nlx2.2(Homeobox)/NPC-Nlx2.2-ChIP-Seq(GSE61673)/Homer	1e-17	50.79%	34.66%
7		Nlx3.1(Homeobox)/LNCaP-Nlx3.1-ChIP-Seq(GSE28264)/Homer	1e-16	52.07%	36.62%

Figure 27: Table of motifs from HOMER

As for de novo motifs, the table is similar to known motifs, with some differences: (Fig. 28)

- *Rank*, *Motif*, *P-value*, *% of targets* and *% of background* are the same as in known motifs.
- *Best match*: it is a display of a known transcription factor which most closely matches the *de novo* motif. Clicking on the text leads to a second page which includes further information about the motif and other matches that HOMER assigns to this motif.

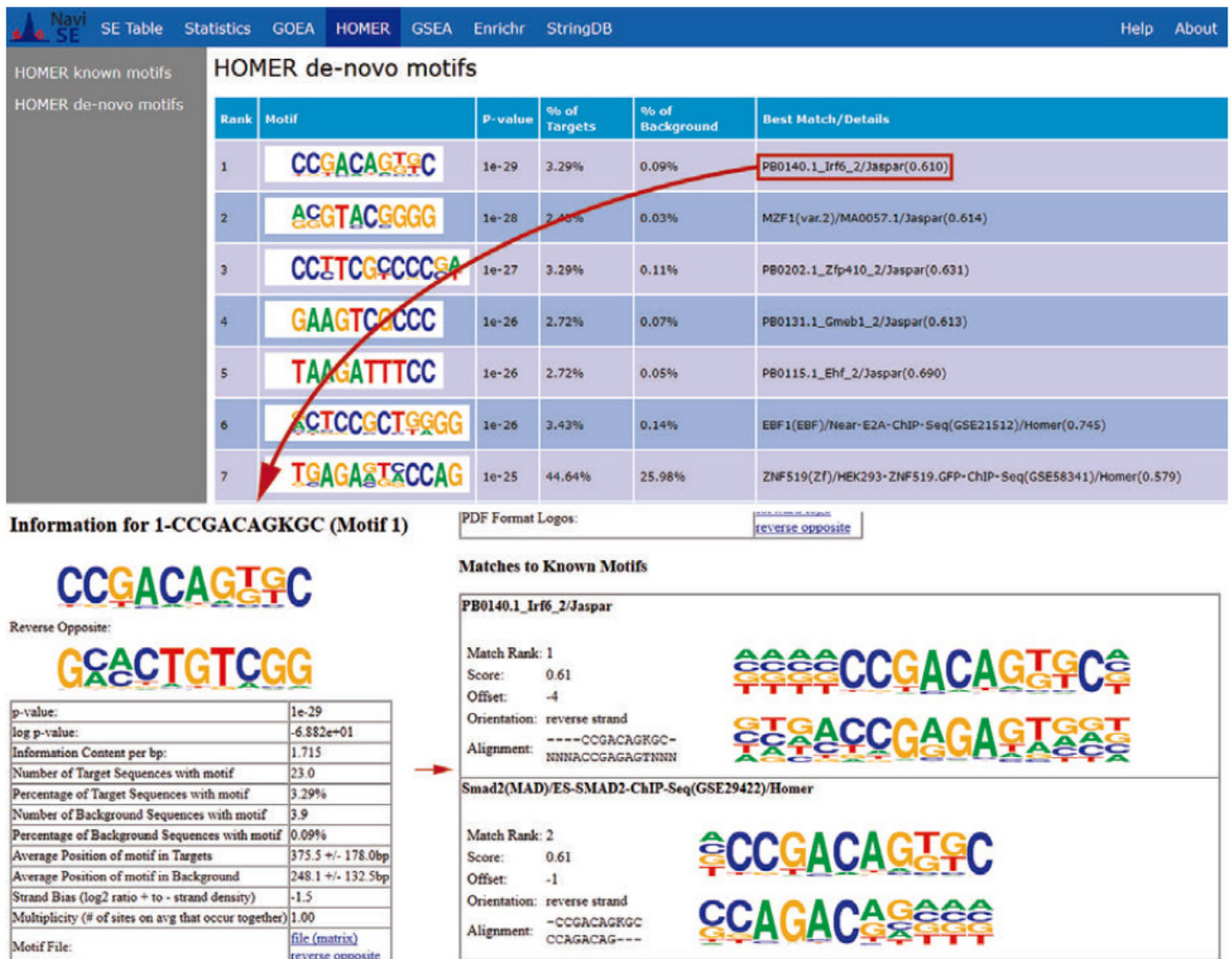


Figure 28: Table of motifs from HOMER

Warning: Analysis of motifs (almost as anything else on NaviSE report) are based on predictions. As HOMER webpage indicates, HOMER results (even more **de novo** result) must be taken **With a grain of salt**. These are orientative results and hence, discovering that the best match of any analysis is the XX transcription factor does not **directly** imply that XX is the main target of superenhancers and plays an essential role for that cellular type in particular. Therefore, it is highly recommendable to take some time and do a thorough comparison between the multiple results NaviSE offers before making erroneous claims. We personally recommend taking into account the [personal tips](#) HOMER offers before launching into analyzing HOMER results

Enrichr results

As it is explained above, Enrichr comprises a number of databases related to transcription factors or genetic regulation (ENCODE/ChEA, JASPAR/TRANSFAC), cell/tissue specification or metabolic pathways (KEGG, Wikipathways, Reactome).

The overall of the page (Fig. 29) contains all the figures first and the tables afterward. Clicking on each name in the barplot will lead to its corresponding term in the table. Each barplot represents the $-\log_{10}(\text{p-value})$ of the corresponding term. Depending on the p-value, there are three possible colors for each bar: gray, if p-value > 0.05, **light colour** if p-value < 0.05 and **dark colour** if p-value < 0.01. Bars are ordered depending on their p-value. Each result is explained below.

TRANSFAC and JASPAR PWMs

TRANSFAC/JASPAR (Fig. 30) contains information about transcription factors related to the superenhancer set in the cell/tissue. The table contains several columns: Term indicates the transcription factor associated with the superenhancer, with positive superenhancers matching to that transcription factor appearing in Genes column. Both Term and Genes contain links to GeneCards of their respective genes. Moreover, other two columns (which also appear in other tables) are Adjusted p-value and Z-score, which are intrinsic values indicating the quality of the match.

ENCODE/ChEA Transcription factors from CHIP-X

ENCODE/ChEA, similar to TRANSFAC/JASPAR, yields a list of transcription factors related to the set of superenhancers (Fig. 31). The distribution of columns is identical to TRANSFAC/JASPAR, with Term and Genes columns containing linkable items to GeneCards page of the corresponding gene.

Gene Atlas

Gene Atlas includes cell types to which there might be some relationship with the genes associated with superenhancers in the sample. Thus, the aim of this table would be to indicate to which tissue/cell type the sample may belong. The table (Fig. 32) contains a Term column, which represents the cell type or tissue, and a Genes column with the genes corresponding to the sample that correlate to the Term. These genes are linked to their respective GeneCards page.

Wikipaths, KEGG Pathways and Reactome

Those three sections contain information about pathways (similar to GOEA or GSEA) whose genes will be present in the superenhancers of the sample. Concerning the content of the columns, they are practically the same as the ones beforehand. We may remark two of the columns: (Fig. 33) The ID

column contains a unique identificative item that is related to the name in Term column. This ID name is linked to its respective pathway webpage (i.e. Reactome, Wikipathways or KEGG pathways). The superenhancer genes related to that Term are located in Genes column, which are linked to their respective GeneCards page.

StringDB results

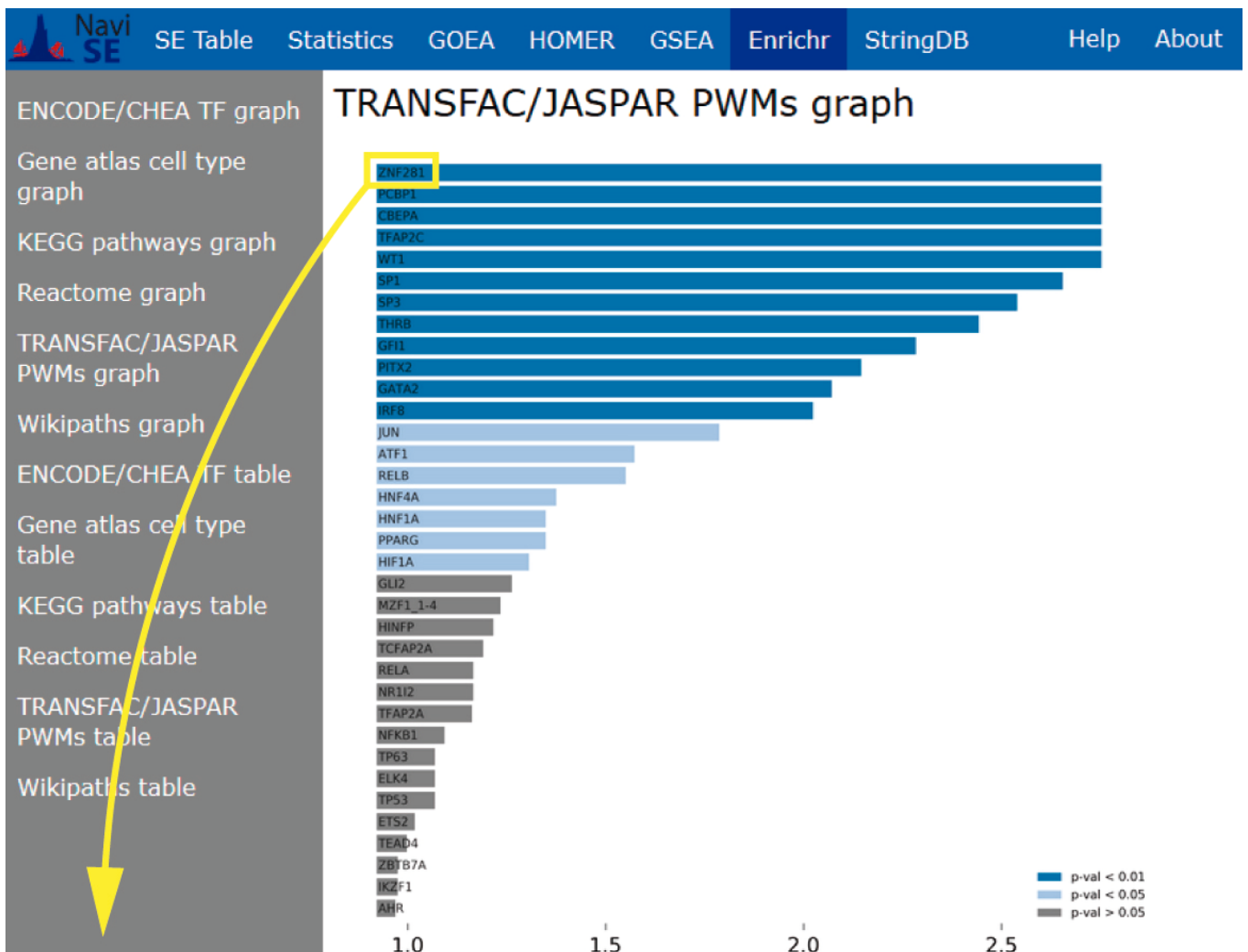
String section contains the protein-protein interaction networks from superenhancers (Fig. 34). There are different *confidence* values determined by how well documented the interaction is, amongst other factors. The current confidence values are 0.4, 0.7, 0.9, 0.95 and 0.99. This range of values allows users to choose the network that best fits their needs, as networks with low *confidence* levels will be overcrowded whereas networks with high *confidence* levels may contain few elements. The links between nodes (genes) are colored in a color code, and the nodes vary in size, as shown in the legend below (Fig. 35).

GSEA results

The *GSEA* section (Fig. 36) includes all the positive analysis for the selected thresholds and signatures. Each signature contains several graphs ordered alphabetically depicting the GSEA curve of the analysis. Generally, the more pronounced and hip-like the curve, the better.

Each graph contains a curve, and below it there is a box with black lines that depict the matches of superenhancer + typical enhancers to the genes corresponding to the gene set from the signature. Below this box there is a graph which shows the value of each position (in this case, the superenhancer score).

Clicking on a graph leads to its corresponding element on a table below, which contains several columns. Focusing on the *SE Genes (Rank)* column, each cell contains genes corresponding to those positive matches, which lead to their respective GeneCards site, and inside parenthesis, there is a number that represents the rank of this gene in the superenhancer list. Clicking on this value will lead to the respective row from *SE Table* section.



To GeneCards

Rank	Term	Adjusted P-value	Z-score	Genes
1	ZNF281	1.767E-03	-1.816	PHC1, SH2B3, SPTA2, ZMYND8, FZD7, TNFRSF1A, DAZL, EEF1A1, TRIM8, GNG4, PODXL, PIM1, ROR1, GYG2, VCL, BCL9L
2	PCBP1	1.767E-03	-1.760	CBX7, HIP1, TSHZ3, FZD7, PXN, MSH6, CDH4, ABHD11, PPP2R4, GNG4, PODXL, CBS, CARM1, KIAA1522, IGF2BP1, PIM1, MYH9, PTMA, VCL, TEAD3
3	CBEPA	1.767E-03	-1.674	ATF7IP, TCF7L2, ZFHX2, ZMYND8, SOX13, ANK2, TRIM8, HIST2H4A, NCOR2, CLEC2A, ERBB2, SERPINH1, MYH9, IGF2BP3, PCDH1, ZNF423, VCL, CTSC, FGFR1
4	TFAP2C	1.767E-03	-1.650	BTBD19, FAM46B, MDH1, IFNGR2, ZMYND8, VASH2, UBE2G1, PWP1, HK1, CYP26A1, ZFP36, PODXL, CARM1, KIAA1522, LMNA, ERBB2, ROR1, PCDH1, FAM60A, TEAD3
5	SP1	2.216E-03	-1.687	CBX7, HIP1, TSHZ3, FZD7, TRIM8, NCOR2, MSH6, ZFP36, ABHD11, PPP2R4, PODXL, CBS, GNG4, CARM1, PIM1, IGF2BP1, IGF2BP3, PTMA, TEAD3
7	WT1	1.767E-03	-1.582	TSHZ3, PXN, CTNND1, TRIM8, CDH4, ABHD11, GYLTL1B, DNAJB6, PPP2R4, PODXL, CBS, GNG4, KIAA1522, ERBB2, PIM1, TLK1, PRSS8, PCDH1, TEAD3, CBX7, ZFHX2, KDM4C, C2ORF57, FZD7, NCOR2, DBNDD1, LIN28A, CARM1, VCL
8	SP3	2.891E-03	-1.673	CBX7, HIP1, TSHZ3, FZD7, PXN, MSH6, CDH4, ZFP36, ABHD11, PPP2R4, PODXL, CBS, GNG4, CARM1, IGF2BP1, PIM1, MYH9, TEAD3
10	THRB	3.606E-03	-1.586	RPL21, SEMA4D, ZC3H4, FZD7, GFPT2, HK1, TRIM8, CYP26A1, PPM1B, ABHD11, NUAQ2, ABLIM2, PPP2R4, CBS, KIAA1522, GNA12, IGF2BP1, MYH9, MFGE8, SERINC5, VCL, FAM60A, FGFR1
11	GFI1	5.201E-03	-1.651	ATF7IP, SPEN, UBE2H, ZFHX2, KDM4C, FAM46B, IFNGR2, PXN, CTNND1, SOX13, ANK2, ACTN4, FABP6, KIAA1522, SERPINH1, MFGE8, VCL, FGFR1
13	GATA2	8.495E-03	-1.594	HIP1, ZMYND8, PIK3CB, IRF2BPL, HK1, SEPT9, PPP2R4, PIM1, TLK1, FAM65B, GYG2, IER2, CTSC, VRTN, LOC100133985, TEAD3, SRRM3, C2ORF57, IFNGR2, VASH2, SOX13, MIR21, ACTW4, TNFRSF1A, EEF1A1, MSH6, PPM1B, CLDN9, KCTD10, TYRO3, MALAT1, MFGE8, PIF1, HIST2H4A, GJC3, NUAQ2, PTPRZ1, DNAJB6, GNG4, CBS, OLFML3, LMNA, ERBB2, MIR1252, SERPINH1, PCDH1, N4BP3, PP1A4F, BCL9L, ATF7IP, CBX7, FAM46B, SEMA4D, FZD7, GFPT2, MIR3914-1, IGFL3, DAZL, IL22RA1, CYP26A1, LIN28B, CLEC2A, LIN28A, IL7, MIR187, CARM1, LOC728743, FGFR1

Figure 29: Enrichr overall window.

TRANSFAC/JASPAR PWMs table **To GeneCards**

Rank	Term	Adjusted P-value	Z-score	Genes
1	ZNF281	1.767E-03	-1.816	PHC1, GMEB, GFF2, ZMYND8, FZD7, TNFRSF1A, DAZL, EEF1A1, TRIM8, GNG4, PODXL, PIM1, ROR1, GYG2, VCL, BCL9L
2	PCBP1	1.767E-03	-1.760	CBX7, HIP1, TSHZ3, FZD7, PXN, MSH6, CDH4, ABHD11, PPP2R4, GNG4, PODXL, CBS, CARM1, KIAA1522, IGF2BP1, PIM1, MYH9, PTMA, VCL, TEAD3
3	CBEPA	1.767E-03	-1.674	ATF7IP, TCF7L2, ZFXZ2, ZMYND8, SOX13, ANK2, TRIM8, HIST2H4A, NCOR2, CLEC2A, ERBB2, SERPINH1, MYH9, IGF2BP3, PCDH1, ZNF423, VCL, CTSC, FGFR1
4	TFAP2C	1.767E-03	-1.650	BTBD19, FAM46B, MDH1, IFNGR2, ZMYND8, VASH2, UBE2G1, PWP1, HK1, CYP26A1, ZFP36, PODXL, CARM1, KIAA1522, LMNA, ERBB2, ROR1, PCDH1, FAM60A, TEAD3
5	SPI	2.216E-03	-1.687	CBX7, HIP1, TSHZ3, FZD7, TRIM8, NCOR2, MSH6, ZFP36, ABHD11, PPP2R4, PODXL, CBS, GNG4, CARM1, PIM1, IGF2BP1, IGF2BP3, PTMA, TEAD3

Figure 30: Transfac/Jaspar PWMs table.

Navise SE Table Statistics GOEA HOMER GSEA Enrichr StringDB Help About

ENCODE/CHEA TF graph
Gene atlas cell type graph
KEGG pathways graph
Reactome graph
TRANSFAC/JASPAR PWMs graph
Wikipaths graph
ENCODE/CHEA TF table
Gene atlas cell type table
KEGG pathways table
Reactome table
TRANSFAC/JASPAR PWMs table
Wikipaths table

ENCODE/CHEA TF table **To GeneCards**

Rank	Term	Adjusted P-value	Z-score	Genes
1	TCF3	1.198E-04	-1.668	CBX7, PHC1, KDM4C, RPL21, ZMYND8, FZD7, ACTH4, PRRC2B, IRF2BP1, MSH6, TRIM8, ABHD11, DNABJ6, PODXL, KIAA1522, PIM1, SERPINH1, TYRO3, BCAT1, PTMA, VRTN, FAM60A, FGFR1
2	SOX2	2.148E-03	-1.718	ATF7IP, CBX7, PHC1, FZD7, MSH6, TRIM8, PPM1B, ABHD11, PTPRZ1, DNABJ6, PODXL, KIAA1522, ROR1, BCAT1, PTMA, SERINC5, VRTN, FGFR1
3	HANOG	2.702E-03	-1.621	CBX7, PHC1, ACTH4, TRIM8, MSH6, ABHD11, PODXL, PIM1, MYH9, ROR1, PRSS8, BCAT1, SERINC5, FAM60A, FGFR1
4	GATA2	0.0248	-1.636	SEMA4D, ZMYND8, IFNGR2, ACTH4, TNFRSF1A, HK1, EEF1A1, SEPT9, KIAA1522, LMNA, GNAI2, PIM1, MYH9, RNF120, IER2
5	RUNX1	0.0248	-1.523	BTBD19, SEMA4D, ZC3H4, ZMYND8, CTNND1, ANK2, PIK3CB, PTPN14, MSH6, TRIM8, SEPT9, LING2A, DNABJ6, KCTD10, CLDN9, PIM1, IGF2BP3, FAM60B, VCL, IER2, BCL9L
6	PPARD	0.0344	-1.548	TRIM8, ACSM6, MYH9, FAM60B, UBE2G1, HALAT1, MFGE8, SUND1P1
7	SALL4	0.1062	-1.480	MSH6, UBE2H, PPP2R4, ZMYND8, SERPINH1, ANK2, BCAT1, IRF2BP1
8	GATA1	0.1436	-1.383	ZMYND8, IFNGR2, HK1, EEF1A1, ZFP36, DNABJ6, KIAA1522, GNAI2, PIM1, TLK1, MYH9, PLEKHO1, RNF150
9	FOSL2	0.2224	-1.467	LINC01011, BTBD19, FAM46B, LMNA, BCL9L
10	TRIM28	0.2364	-1.395	MSH6, PODXL, FZD7, PIM1, PTMA
11	PPARG	0.2364	-1.355	SEPT9, C2ORF57, ZMYND8, CTNND1, LRRK1, CD9, MYH9, ACTH4, PTMA
12	MYC	0.2837	-1.298	HIST2H4A, CBX7, RPL21, DNABJ6, IGF2BP1, PWP1, PTMA, IER2, FAM60A
13	ZBTB7A	0.2812	-1.261	TFH2, LINC01011, ZC3H4, TRIM8, PPP2R4, CBS, KIAA1522, IGF2BP1, PIM1, IGF2BP3, IER2, TEAD3, BCL9L, SRRM3, IFNGR2, MSH6, EEF1A1, NCOR2, DBND1, CARM1, PTPN4, LOC728743, PTMA, VCL, FAM60A

Figure 31: ENCODE/CHEA TF table.

Gene atlas cell type table **To GeneCards**

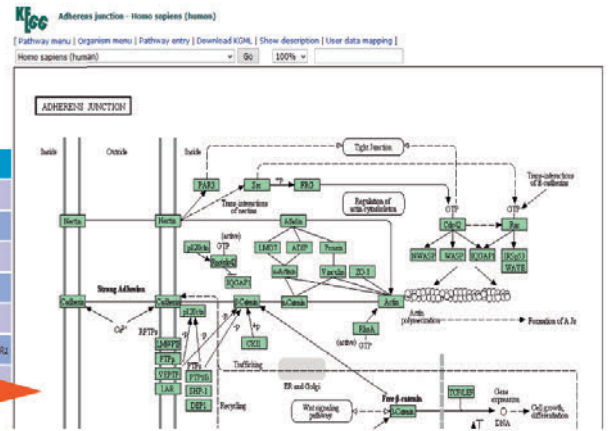
Rank	Term	Adjusted P-value	Z-score	Genes
1	Adipocyte	1.0000	-1.494	OLFML3, ROR1, MFGE8, GYG2
2	SmoothMuscle	1.0000	-1.677	GPR176, GFF2, MYO6, SERPINH1, BCAT1, FGF2
3	CD19+_BCells(neg._sel.)	1.0000	-1.814	GYLTL1B, KDM4C, IL7, TLK1, PLEKHO1, PWP1
4	Placenta	1.0000	-1.526	ERBB2, SOX13, IGF2BP3, PRSS8, PCDH1, TEAD3
5	Prostate	1.0000	-1.439	ABHD11, ROR1, SERINC5

Figure 32: Gene atlas cell type table.

As for the rest of the columns, which appear in more detail [here](#), ES and NES are the Enrichment Score and the Normalised Enrichment Score. ES reflects the degree to which a gene set is over-represented at the top or bottom of a ranked list of genes. The ES is the maximum deviation from zero encountered in walking the list. A positive ES indicates gene set enrichment at the top of the

KEGG pathways graph

Adherens junction
Proteoglycans in cancer
Regulation of actin cytoskeleton
Leukocyte transendothelial migration
Central carbon metabolism in cancer
Pathways in cancer



KEGG pathways table

Rank	ID	Term	Adjusted p-value	Z-score	Genes
1	hsa04520	Adherens junction	5.479E-03	-1.871	TOPYLL, CTNND1, ERBB2, ACTN4, VCL, FOPR1
2	hsa05205	Proteoglycans in cancer	0.0156	-1.907	FZD7, PKN, ERBB2, ANK2, MIR21, PIK3CB, FGF2, FGFRL1
3	hsa04810	Regulation of actin cytoskeleton	0.0156	-1.871	PKN, GNAI2, MYH9, ACTN4, PIK3CB, FGF2, VCL, FOPR1
4	hsa04670	Leukocyte transendothelial migration	0.0156	-1.649	CLDN5, CTNND1, PKN, ACTN4, PIK3CB, VCL
5	hsa05230	Central carbon metabolism in cancer	0.0705	-1.752	ERBB2, PIK3CB, HK1, FOPR1
6	hsa05200	Pathways in cancer	0.0924	-1.950	HGH, TCF7L2, GNAQ, FZD7, ERBB2, GNAI2, PIK3CB, FGF2, FGFRL1
7	hsa05215	Prostate cancer	0.1070	-1.774	ERBB2, ERBB3, ERBB4, FOPR1

To GeneBank

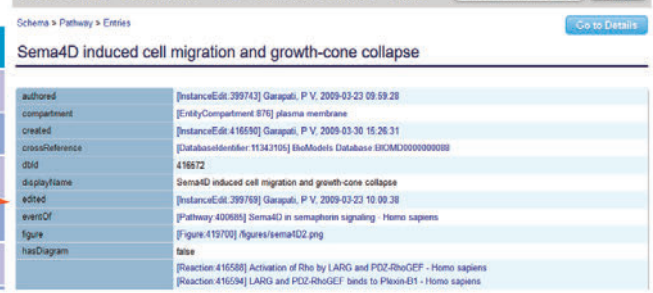
Wikipaths graph

Signaling Pathways in Glioblastoma
Regulation of Actin Cytoskeleton
Primary Focal Segmental Glomerulosclerosis FSGS
MicroRNAs in cardiomyocyte hypertrophy



Reactome table

Rank	ID	Term	Adjusted P-value	Z-score	Genes
1	R-HSA-416572	Sema4D induced cell migration and growth-cone collapse	0.1097	-2.162	SEMA4D, ERBB2, MYH9
2	R-HSA-400695	Sema4D in semaphorin signaling	0.1097	-2.120	SEMA4D, PIK3D, MYH9
3	R-HSA-180292	GAB1 signalosome	0.1257	-2.181	PKN, ERBB2, PIK3CB, FGF2, FOPR1
4	R-HSA-3371511	HSF1 activation	0.1097	-2.036	HSF1A1, DNAH8, SEPPOR2
5	R-HSA-1500931	Cell-Cell communication	0.1097	-2.023	CDH4, CLDN8, CTNND1, PKN, ACTN4, PIK3CB



Reactome graph

Sema4D induced cell migration and growth-cone collapse
Sema4D in semaphorin signaling
HSF1 activation
Cell-Cell communication



Wikipaths table

Rank	ID	Term	Adjusted P-value	Z-score	Genes
1	WP2261	Signaling Pathways in Glioblastoma	0.2910	-2.268	HGH, ERBB2, MIR21, PIK3CB, FGF2
2	WP51	Regulation of Actin Cytoskeleton	0.2910	-2.099	PKN, GNAI2, PIK3CB, FGF2, VCL, FOPR1
4	WP2572	Primary Focal Segmental Glomerulosclerosis FSGS	0.3222	-1.831	PODIL, MYH9, ACTN4, VCL
5	WP1544	MicroRNAs in cardiomyocyte hypertrophy	0.5482	-1.931	MIR22A, MIR21, PIK3CB, FGF2
6	WP399	Wnt Signaling Pathway and Pluripotency	0.5482	-1.944	PCP4, CTNND1, FZD7, FZD4, FZD3, FZD2, FZD1, FZD5, FZD6, FZD7L1, FZD7L2, FZD7L3, FZD7L4, FZD7L5, FZD7L6, FZD7L7, FZD7L8, FZD7L9, FZD7L10, FZD7L11, FZD7L12, FZD7L13, FZD7L14, FZD7L15, FZD7L16, FZD7L17, FZD7L18, FZD7L19, FZD7L20, FZD7L21, FZD7L22, FZD7L23, FZD7L24, FZD7L25, FZD7L26, FZD7L27, FZD7L28, FZD7L29, FZD7L30, FZD7L31, FZD7L32, FZD7L33, FZD7L34, FZD7L35, FZD7L36, FZD7L37, FZD7L38, FZD7L39, FZD7L40, FZD7L41, FZD7L42, FZD7L43, FZD7L44, FZD7L45, FZD7L46, FZD7L47, FZD7L48, FZD7L49, FZD7L50, FZD7L51, FZD7L52, FZD7L53, FZD7L54, FZD7L55, FZD7L56, FZD7L57, FZD7L58, FZD7L59, FZD7L60, FZD7L61, FZD7L62, FZD7L63, FZD7L64, FZD7L65, FZD7L66, FZD7L67, FZD7L68, FZD7L69, FZD7L70, FZD7L71, FZD7L72, FZD7L73, FZD7L74, FZD7L75, FZD7L76, FZD7L77, FZD7L78, FZD7L79, FZD7L80, FZD7L81, FZD7L82, FZD7L83, FZD7L84, FZD7L85, FZD7L86, FZD7L87, FZD7L88, FZD7L89, FZD7L90, FZD7L91, FZD7L92, FZD7L93, FZD7L94, FZD7L95, FZD7L96, FZD7L97, FZD7L98, FZD7L99, FZD7L100
10	WP2406	Cardiac Progenitor Differentiation	0.5482	-1.790	LINC8B, LINC8A, FGF2

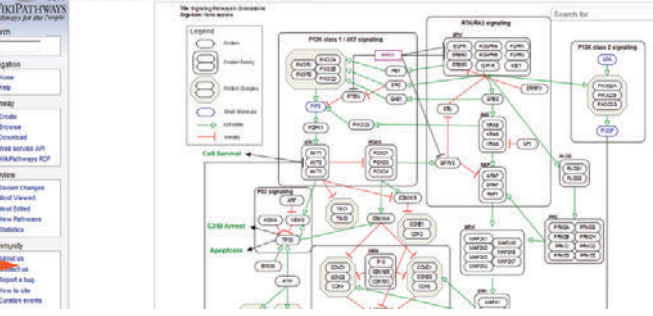


Figure 33: Metabolism pathways tables and main targets.

ranked list; a negative ES indicates gene set enrichment at the bottom of the ranked list. In our case, we are not going to find negative ES. The normalized enrichment score (NES) is the primary statistic for examining gene set enrichment results. By normalizing the enrichment score, GSEA accounts for differences in gene set size and in correlations between gene sets and the expression dataset; therefore, the normalized enrichment scores (NES) can be used to compare analysis results across

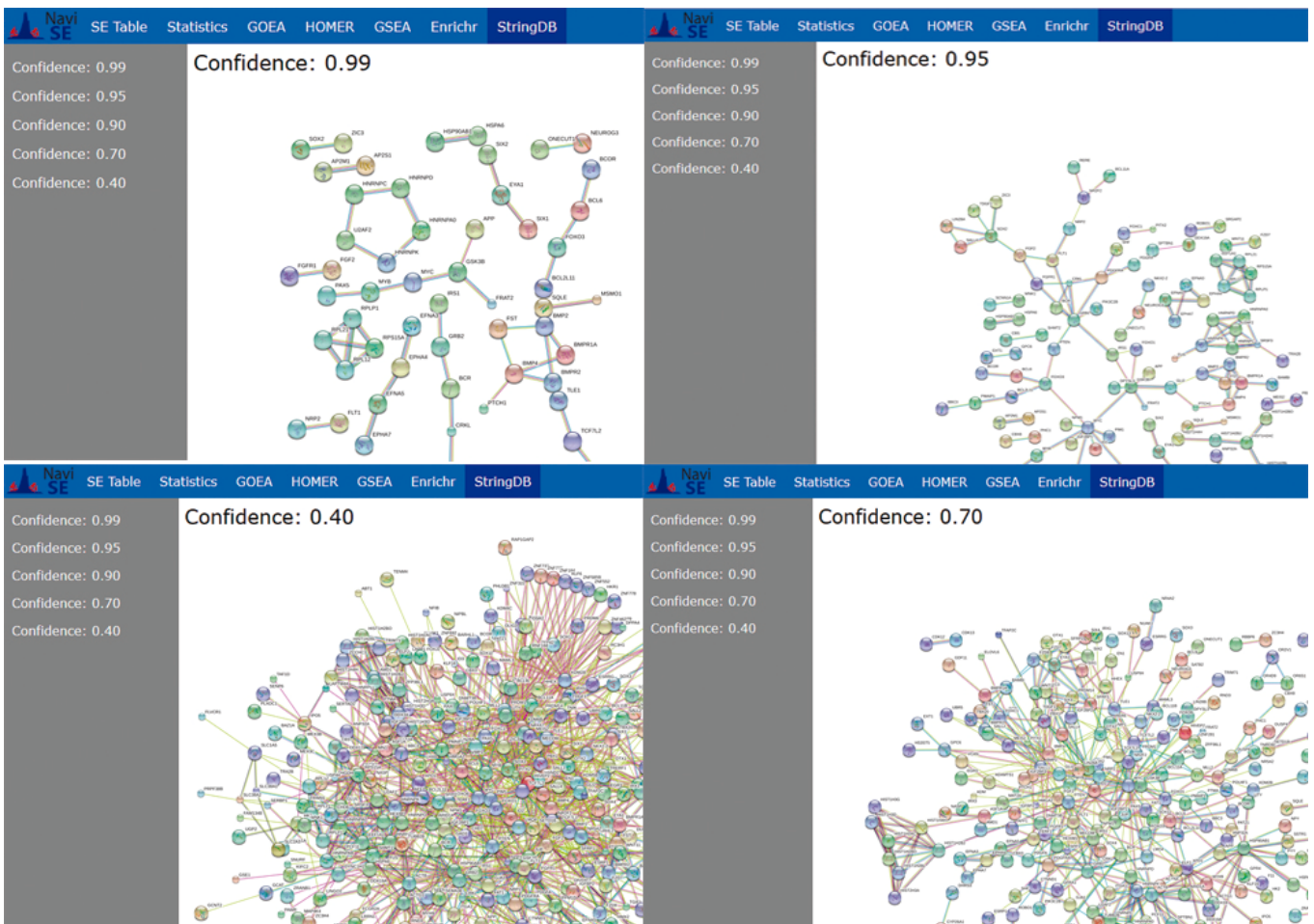


Figure 34: String window.

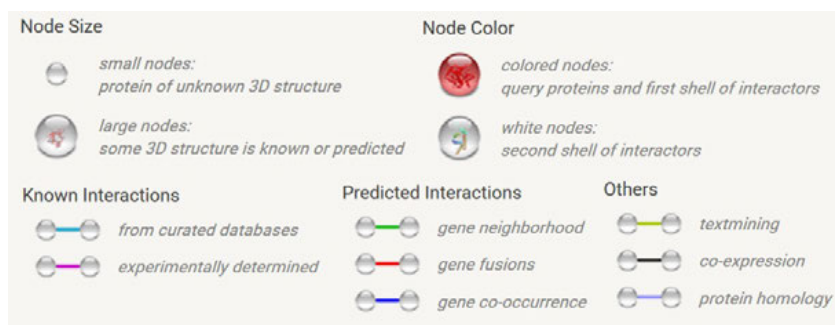


Figure 35: String legend.

gene sets.

The **FDR** is the estimated probability that a gene set with a given NES represents a false positive finding. Thus, the smaller the FDR the better. The **nominal p value** estimates the statistical significance of the enrichment score for a single gene set. Finally, the **-log₁₀(Ratio p-value)** corresponds to the hypergeometric test between the number of matches between SE and TE, and the number of SE and TE (with or without matched) that our sample contains. This p-value should be indicative of

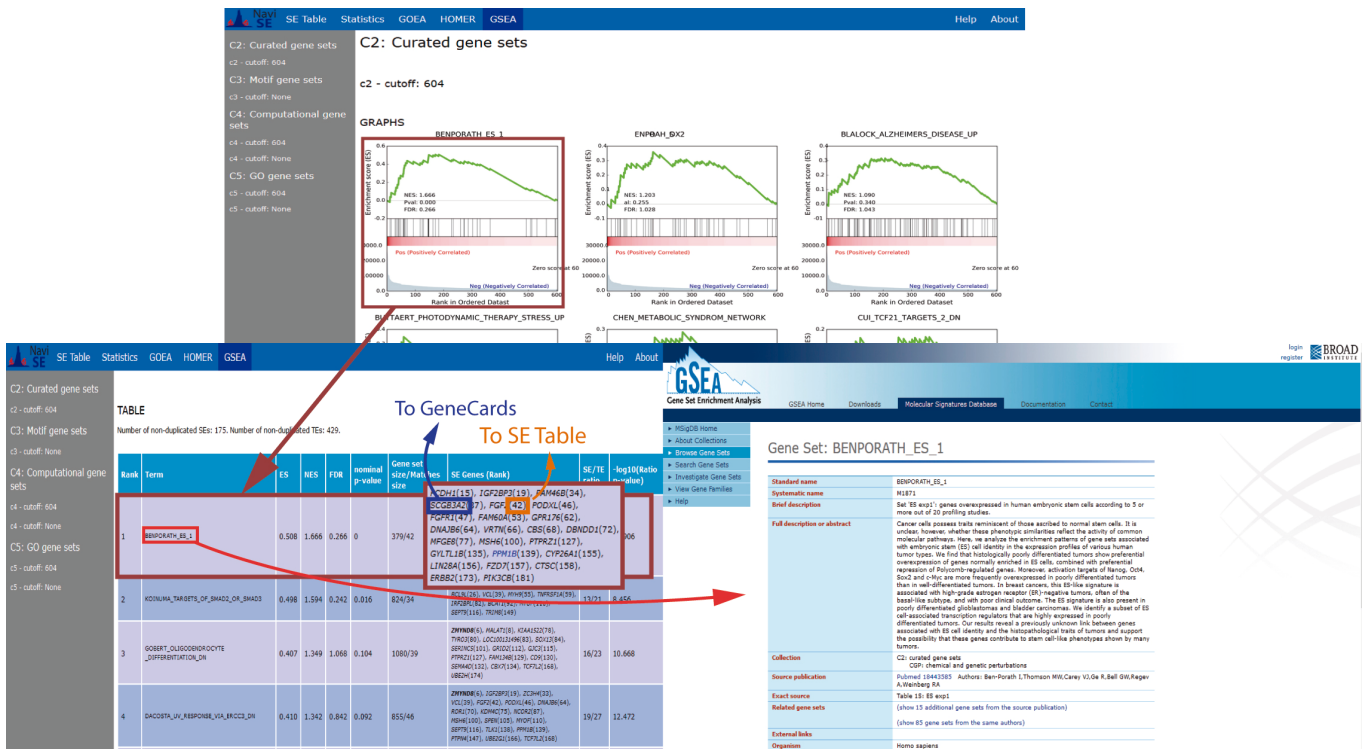


Figure 36: GSEA window.

how enriched in SE matches against TE matches the sample is.

Thus, the user should be able to discern which graphs are really representative based in FDR, nominal p-value, distribution of the curve and NES, from those graphs which also appear but are not fully reliable or statistically significant.

