GSCA: Gene Set Context Analysis

Zhicheng Ji, Hongkai Ji

September 14, 2026

1 Introduction

Although techniques of chromatin immunoprecipitation coupled with high throughput sequencing (ChIP-seq) or tiling array hybridization (ChIP-chip) have came into being for a long time, it still remains difficult to generate a quality ChIPx (i.e., ChIP-seq or ChIP-chip) data set due to the tremendous amount of effort required to develop effective antibodies and efficient protocols. Especially with recent cuts in research fundings, most labs are unable to easily obtain ChIPx data in more than a handful of biological contexts. Thus, standard ChIPx analyses primarily focus on analyzing data from one experiment, and the discoveries are restricted to a specific biological context. We propose to enrich this existing data analysis paradigm by developing a novel approach, GSCA, which superimposes ChIPx data on large amounts of publicly available human and mouse gene expression data containing a diverse collection of cell types, tissues, and disease conditions to discover new biological contexts with potential geneset activity patterns. GSCA could also serve as an informative guide for biologists to prescreen interested biological contexts when designing their experiments.

2 Overview

The purpose of GSCA is to predict the biological contexts, defined as the cell or disease type and associated treatment or condition, in which a certain geneset activity pattern exhibits. GSCA accomplishes this by first requiring the users to specify a number of genesets with activated (positive) and repressed (negative) genes defined from experimental data from one or more cell types. Users are also required to specify a particular geneset activity pattern they want to study. Given the genesets, GSCA will then search for biological contexts that are significantly enriched with the specific geneset activity pattern by examining the activity value of each given geneset across all of the biological contexts in the gene expression compendium. Finally, a complete report including result tables and plots will be generated.

3 GSCA analysis

Two key things are required to perform a GSCA analysis: (1) genesets with activated and repressed genes (2) geneset activity pattern of interest.

(1) Genesets with activated and repressed genes - users can give any genesets they wish to study, but normally these genesets come from experimental data. Activated gene means that increases in expression of the gene also increases the overall activity of the whole geneset, while increases in expression of the repressed genes will decrease the overall activity of the whole geneset.

(2) Geneset activity pattern of interest - users need to specify the geneset activity pattern of interest. For each geneset, users should define either high or low activity to be searched by GSCA, a cutoff type and a cutoff value. For example, if the pattern is set to be high in all given genesets, then GSCA will search for samples whose geneset activity values are all above the respective cutoffs calculated in each geneset.

After providing the required input, GSCA identifies biological contexts enriched with the geneset activity pattern of interest by counting the number of samples that fall within and outside of the defined geneset activity region for each biological context in the gene expression compendium. Fisher's exact test is then used to calculate the probability of association between the geneset activity pattern with each biological context to determine which contexts are significantly enriched with given geneset activity pattern. See the reference below for more details on the GSCA algorithm.

Besides the standard R functions provided in GSCA package, users are recommended to use the interactive GSCA user interface built using R shiny. The GSCA UI includes nearly all GSCA standard R functions while providing a easier way and more powerful options to do the analyses and generate outputs. Please check the help page of GSCAui() for more details.

4 GSCA function

GSCA analysis is performed using the GSCA function, which requires the following input arguments:

(1) `genedata`, A data.frame with three columns specifying the input genesets. Each row specifies an activated or repressed gene in a geneset. First column: character value of geneset name specified by the user, could be any name easy to remember e.g. GS1,GS2,...; Second column: numeric value of Entrez GeneID of the gene; Third column: numeric value of 1,-1 indicating whether gene is activated or repressed. 1 for activated gene and -1 for repressed gene. Here, activated gene means that increases in expression of the gene also increases the overall activity of the whole geneset, while increases in expression of the repressed genes will decrease the overall activity of the whole geneset.

(2) `pattern`, A data.frame with four columns indicating the activity patterns corresponding to the given genedata. Each row specifies activity pattern for one geneset. First column: character value of the same geneset name used in genedata, each geneset name in genedata should appear exactly once in this column. Second column: character value of whether high or low activity of the whole geneset is interested. "High" stands for high activity and "Low" stands for low activity. Third column: character value of which cutoff type is going to be used. 3 cutoff types can be specified: "Norm", "Quantile", or "Exprs". If cutoff type is "Norm", then the fourth column should be specified as p-value between 0 and 1, where the geneset expression cutoff will correspond to the specified p-value

(one-sided) based on a fitted normal distribution; If cutoff type is "Quantile", then the fourth column should be specified as a desired quantile between 0 and 1, where the geneset expression cutoff will correspond to the specified quantile. Finally, if cutoff type is "Exprs", the geneset expression cutoff will be equal to the value given in the fourth column. Fourth column: numeric value of cutoff value based on different cutoff types specified in the third column.

(3) `chipdata`, Compendium data in which the analysis is performed. Two data sets are possible: 'moe4302' or 'hgu133a'. Based on the input, one of two currently available compendiums of publicly available gene expression profiles, from GPL96 for human or from GPL1261 for mouse, will be loaded. The gene expression compendium data is downloaded from NCBI GEO (Barret et al. 2007).

(4) `Pval.co`, Significance cutoff for reported active biological contexts.

(5) `directory`, Either null or a character value giving a directory path. If directory is not null, then additional follow-up GSCA analyses will be performed and stored in the folder specified by directory. If directory is null then no additional follow-up GSCA analyses will be performed.

5 GSCA Example

Here, we illustrate an example of how to use the GSCA function to produce GSCA active biological context predictions. Suppose we are interested in studying Oct4 regulation in mouse embryonic stem cells (ESCs) and already have ChIP-seq data for Oct4 in ESCs and gene expression data before and after RNAi knockdown of Oct4 in ESCs. First, we process ChIP-seq data using CisGenome (or other viable methods) to obtain a list of predicted Oct4-bound target genes in ESCs. Then, we analyze the gene expression data using RMA and limma (or other viable preprocessing and analysis methods) to obtain a list of differentially expressed genes after RNAi knockdown. Next, we combine the TF-bound genes and differentially expressed genes to obtain a set of Oct4 target genes in ESCs. To be specific, positive target genes are genes that are TF-bound and increases in expression when the TF expression increases, and negative target genes are genes that are TF-bound and decrease in expression when the TF expression decreases. This has already been done previously and has been stored as a list in the GSCA package. The ChIP-seq data is obtained from GSE11431 and the Oct4-knockdown data is obtained from GSE4189.

We are now ready to load the input data.

```
> library(GSCA)
> data(Oct4ESC_TG)
> head(Oct4ESC_TG[[1]]) ##Show some positive target genes of Oct4

[1] "100678" "106298" "14609"  "12468"  "16765"  "21849"

> head(Oct4ESC_TG[[2]]) ##Show some negative target genes of Oct4

[1] "246703" "15441"  "70579"  "20333"  "83669"  "245688"
```

Before running the GSCA function, we should first construct the genedata and specify the geneset acitivty pattern which are both required. First we build

the genedata. We specify the Entrez GeneID of the TF-of-interest (Entrez GeneID of Oct4: 18999) and the vector of EntrezIDs for the positive (activated) and negative (repressed) Oct4 target genes:

```
> activenum <- length(Oct4ESC_TG[[1]])
> repressnum <- length(Oct4ESC_TG[[2]])
> Octgenedata <- data.frame(
+       gsname=c("TF",rep("TG",activenum+repressnum)),
+       gene=c(18999,Oct4ESC_TG[[1]],Oct4ESC_TG[[2]]),
+       weight=c(rep(1,1+activenum),rep(-1,repressnum)),
+       stringsAsFactors=FALSE)
```

Second we build the geneset activity pattern. In this analysis, we are interested in biological contexts in which the expression of the TF and the activity of its target genes are both high, and define the high activity region as the TF and TG cutoff correspond to a one-sided palue based on fitted normal distributions.

```
> Octpattern <- data.frame(
+       gsname=c("TF","TG"),
+       acttype="High",
+       cotype="Norm",
+       cutoff=0.1,
+       stringsAsFactors=FALSE)
```

We are now ready to run the GSCA function. We specify chipdata as "moe4302" and Pvalue cutoff as default: 0.05.

```
> displayoct <- Octoutput <- GSCA(Octgenedata,Octpattern,"moe4302",Pval.co=0.05,directory=
> displayoct[[1]]$SampleType <- substr(displayoct[[1]]$SampleType,1,25)
> head(displayoct[[1]]) ## Partial results of the ranking table
```

```
  Rank Active Total FoldChange Adj.Pvalue               SampleType
1    1     20    20     19.904   2.89e-24 inner_cell_mass_cell:gene
2    2     18    20     17.919   2.46e-19 single_cell_from_blimpko_
3    3     16    17     18.589   1.11e-17 single_cell_from_lineager
4    4     13    13     19.432   7.09e-15 embryonic_stem_cells:norm
5    5     12    12     19.323   1.54e-13 embryoid_bodies:r1,_diffe
6    6     12    12     19.323   1.54e-13 embryonic_stem_cells:r1,_
                                                    ExperimentID
1                                                GSE4307;GSE4309
2                                                       GSE11128
3                                                       GSE11128
4 GSE9954;GSE10476;GSE10573;GSE10553;GSE10610;GSE10776;GSE10806
5                                                        GSE9563
6                                                        GSE9563
```

The first item in the output contains the ranking table of biological contexts significantly enriched with the regulatory pattern that both TF and its target genes have high activities. Since the Pval.co is set to 0.05, only biological contexts with adjusted p-values less than 0.05 are reported. It is important to note that the TF expression and TG activity cutoffs as set by the third and fourth columns of pattern are completely flexible; users are free to determine

how 'high' expression of activity needs to be based on how stringent they would like to be with the resulting predictions (e.g. higher TF and TG cutoff will result in more stringent predictions). Real data tests show that different TFs can behave very differently so there is no clear optimal cutoff for all TFs.

The output also contains the TF expression and TG activity scores for each sample in the compendium, the number of total target genes and target genes that did not have expression measurements on the compendium, and the value of the TF expression and TG activity cutoff.

If the user would like to visualize the GSCA results, we also provide a function to quickly plot the output.

```
> GSCAplot(Octoutput,N=5,plotfile=NULL,Title="GSCA plot of Oct4 in ESC")
```

The plot depicts the TF expression and TG activity scores for each sample in the compendium and highlights in color the samples from the top-ranked enriched biological contexts. The dashed lines indicate the TF expression and TG activity cutoffs to visualize the defined regulatory region of interest. Since most names for the biological contexts are rather long, only the first portion (maximum of 25 characters) of the name is shown in the legend. Simply refer to the output table for the full sample type name.

The first argument to GSCAplot is the direct output from the GSCA function. The second argument, N, specifies the number of top-ranked enriched biological contexts to highlight (with a maximum of 5). The third argument, plotfile, specifies where the plot is saved. If plotfile is left as NULL, then the plot will not be saved but directly shown in R. The fourth argument is Title, which is the title of the plot. GSCAplot function is provided solely for convenience. It is designed to quickly plot the output of GSCA, and leaves no options to customize the resulting plot. If users would like to change the way the plot looks, users can simply launch GSCA UI using the function `GSCAui`.

6 GSCA further exploration

After the initial GSCA analysis, users may want to explore the predicted contexts in more detail. GSCA package contains two functions `tabSearch` and `GSCAeda` which are designed for this purpose. `tabSearch` is used to search in the human or mouse compendium for samples related to a set of keywords of interest. Then the contexts recovered by `tabSearch` are inputted into `GSCAeda` along with the initial GSCA inputs: genedata, pattern, etc. to analyze the set of inputted contexts for differences in geneset activities. Specifically, `GSCAeda` will calcualte the mean and standard deviation of the geneset activity values. To better visualize the data, a boxplot showing activities of each geneset will also be generated . Next, `GSCAeda` will perform t-tests comparing the mean geneset activity values in each geneset for all pair-wise combinations among the inputted contexts, report the results in a table and plot them in two heatmaps showing the t-statistics and p-values. In addition, the usual GSCA tests of enrichment of the geneset activity pattern of interest will be reported. If outputdir argument is not null, all plots will be outputted to the file path specified by the user in pdf format and all data used to construct the plots will be outputted to the same file path in csv format. The raw geneset activity values of each sample

for each inputted context will also be reported in a separate csv file, so users can perform additional statistical analyses.

For example, we can load in STAT1 target genes defined from ChIP-seq and literature.

```
> data(STAT1_TG) ### Note, only activated (+) STAT1 target genes were found
```

Then we construct genedata and pattern required by GSCA.

```
> Statgenenum <- length(STAT1_TG)
> Statgenedata <- data.frame(gsname=c("TF",rep("TG",Statgenenum)),gene=c(6772,STAT1_TG),we
> Statpattern <- data.frame(gsname=c("TF","TG"),acttype="High",cotype="Norm",cutoff=0.1,st
```

Now we can perform an initial GSCA analysis.

```
> Statoutput <- GSCA(Statgenedata,Statpattern,"hgu133a",Pval.co=0.05,directory=NULL)
> head(Statoutput[[1]])
```

```
  Rank Active Total FoldChange Adj.Pvalue               SampleType
1    1     25    60     11.009   9.06e-18  pbmc:hepatitis_c,_day_1
2    2     24    59     10.746   1.08e-16  pbmc:hepatitis_c,_day_2
3    3     26   101      6.847   1.47e-12                blood:sle
4    4     18    52      9.128   1.15e-10 pbmc:hepatitis_c,_day_28
5    5     17    60      7.492   2.17e-08  pbmc:hepatitis_c,_day_7
6    6     10    18     14.170   1.05e-07          pbmc:influenzaa
                ExperimentID
1                     GSE7123
2                     GSE7123
3 GSE11907;GSE11908;GSE11909
4                     GSE7123
5                     GSE7123
6                     GSE6269
```

As we can see, many of the significant predictions come from hepatitis-C infected PBMCs in experiment GSE7123. To further see if there are additional biological insights we can make specific to STAT1 functional activity in hepatitis-C infected PBMCs, we can compare the TF expression and TG activity values for all contexts in GSE7123 using GSCAeda. We use the experiment ID to search in this case because it is likely that the experiment will contain more relavent contexts (possibly not significant ones) to the hepatitis-C infected PBMCs (e.g. PBMCs:healthy).

We first search for all contexts in GSE7123.

```
> GSE7123out <- tabSearch("GSE7123","hgu133a")
> GSE7123out
```

```
  ExperimentID               SampleType SampleCount
1      GSE7123  pbmc:hepatitis_c,_day_0          59
2      GSE7123  pbmc:hepatitis_c,_day_1          60
3      GSE7123  pbmc:hepatitis_c,_day_2          59
4      GSE7123  pbmc:hepatitis_c,_day_7          60
5      GSE7123 pbmc:hepatitis_c,_day_14          54
6      GSE7123 pbmc:hepatitis_c,_day_28          52
```

As we can see, PBMCs:healthy context is also included in this experiment, which was not reported as significant in the initial GSCA analysis results.

Then we run GSCAeda to perform follow-up analysis on the contexts in GSE7123.

```
> GSE7123followup <- GSCAeda(Statgenedata,Statpattern,"hgu133a",GSE7123out,Pval.co=0.05,Or

'data.frame':        344 obs. of  5 variables:
 $ SampleID    : chr  "GSM171172" "GSM171173" "GSM171174" "GSM171175" ...
 $ ExperimentID: chr  "GSE7123" "GSE7123" "GSE7123" "GSE7123" ...
 $ SampleType  : chr  "pbmc:hepatitis_c,_day_0" "pbmc:hepatitis_c,_day_1" "pbmc:hepatitis_
 $ TF          : num  1.05 2.642 5.366 -0.629 -1.304 ...
 $ TG          : num  4.65 4.28 7.06 6.01 4.49 ...
NULL

> GSE7123followup$Tstats

$TF
                        pbmc:hepatitis_c,_day_1 pbmc:hepatitis_c,_day_2
pbmc:hepatitis_c,_day_1              0.00000000             -0.02985787
pbmc:hepatitis_c,_day_2              0.02985787              0.00000000
pbmc:hepatitis_c,_day_28            -1.35220977             -1.39911043
pbmc:hepatitis_c,_day_7             -2.00947632             -2.06836215
pbmc:hepatitis_c,_day_14            -2.18712604             -2.24533411
pbmc:hepatitis_c,_day_0             -4.59227347             -4.70579193
                        pbmc:hepatitis_c,_day_28 pbmc:hepatitis_c,_day_7
pbmc:hepatitis_c,_day_1                1.3522098               2.0094763
pbmc:hepatitis_c,_day_2                1.3991104               2.0683622
pbmc:hepatitis_c,_day_28               0.0000000               0.6200464
pbmc:hepatitis_c,_day_7               -0.6200464               0.0000000
pbmc:hepatitis_c,_day_14              -0.8553939              -0.2753609
pbmc:hepatitis_c,_day_0               -3.2315680              -2.7663650
                        pbmc:hepatitis_c,_day_14 pbmc:hepatitis_c,_day_0
pbmc:hepatitis_c,_day_1                2.1871260                4.592273
pbmc:hepatitis_c,_day_2                2.2453341                4.705792
pbmc:hepatitis_c,_day_28               0.8553939                3.231568
pbmc:hepatitis_c,_day_7                0.2753609                2.766365
pbmc:hepatitis_c,_day_14               0.0000000                2.285384
pbmc:hepatitis_c,_day_0               -2.2853835                0.000000

$TG
                        pbmc:hepatitis_c,_day_1 pbmc:hepatitis_c,_day_2
pbmc:hepatitis_c,_day_1                0.000000                2.410017
pbmc:hepatitis_c,_day_2               -2.410017                0.000000
pbmc:hepatitis_c,_day_28             -10.038888               -7.846065
pbmc:hepatitis_c,_day_7               -8.860577               -6.783606
pbmc:hepatitis_c,_day_14              -9.217474               -6.953762
pbmc:hepatitis_c,_day_0              -22.187324              -19.924067
                        pbmc:hepatitis_c,_day_28 pbmc:hepatitis_c,_day_7
pbmc:hepatitis_c,_day_1                10.0388884               8.8605770
pbmc:hepatitis_c,_day_2                 7.8460649               6.7836059
```

```
pbmc:hepatitis_c,_day_28                0.0000000              -0.5268211
pbmc:hepatitis_c,_day_7                 0.5268211               0.0000000
pbmc:hepatitis_c,_day_14                1.0712620               0.4623230
pbmc:hepatitis_c,_day_0               -10.1024385              -9.8610977
                        pbmc:hepatitis_c,_day_14 pbmc:hepatitis_c,_day_0
pbmc:hepatitis_c,_day_1                  9.217474               22.187324
pbmc:hepatitis_c,_day_2                  6.953762               19.924067
pbmc:hepatitis_c,_day_28                -1.071262               10.102438
pbmc:hepatitis_c,_day_7                 -0.462323                9.861098
pbmc:hepatitis_c,_day_14                 0.000000               11.742543
pbmc:hepatitis_c,_day_0                -11.742543                0.000000
```

> GSE7123followup$Pval

```
$TF
                        pbmc:hepatitis_c,_day_1 pbmc:hepatitis_c,_day_2
pbmc:hepatitis_c,_day_1            1.000000e+00            9.762313e-01
pbmc:hepatitis_c,_day_2            9.762313e-01            1.000000e+00
pbmc:hepatitis_c,_day_28           1.790850e-01            1.646205e-01
pbmc:hepatitis_c,_day_7            4.689329e-02            4.091923e-02
pbmc:hepatitis_c,_day_14           3.083342e-02            2.674354e-02
pbmc:hepatitis_c,_day_0            1.454548e-05            9.398318e-06
                        pbmc:hepatitis_c,_day_28 pbmc:hepatitis_c,_day_7
pbmc:hepatitis_c,_day_1              0.179084967             0.046893287
pbmc:hepatitis_c,_day_2              0.164620470             0.040919226
pbmc:hepatitis_c,_day_28             1.000000000             0.536562431
pbmc:hepatitis_c,_day_7              0.536562431             1.000000000
pbmc:hepatitis_c,_day_14             0.394308363             0.783555770
pbmc:hepatitis_c,_day_0              0.001773984             0.006741822
                        pbmc:hepatitis_c,_day_14 pbmc:hepatitis_c,_day_0
pbmc:hepatitis_c,_day_1               0.03083342            1.454548e-05
pbmc:hepatitis_c,_day_2               0.02674354            9.398318e-06
pbmc:hepatitis_c,_day_28              0.39430836            1.773984e-03
pbmc:hepatitis_c,_day_7               0.78355577            6.741822e-03
pbmc:hepatitis_c,_day_14              1.00000000            2.470850e-02
pbmc:hepatitis_c,_day_0               0.02470850            1.000000e+00

$TG
                        pbmc:hepatitis_c,_day_1 pbmc:hepatitis_c,_day_2
pbmc:hepatitis_c,_day_1            1.000000e+00            1.751115e-02
pbmc:hepatitis_c,_day_2            1.751115e-02            1.000000e+00
pbmc:hepatitis_c,_day_28           3.511239e-17            3.429334e-12
pbmc:hepatitis_c,_day_7            1.134613e-14            5.584289e-10
pbmc:hepatitis_c,_day_14           2.127205e-15            2.618657e-10
pbmc:hepatitis_c,_day_0            3.169190e-41            2.123144e-37
                        pbmc:hepatitis_c,_day_28 pbmc:hepatitis_c,_day_7
pbmc:hepatitis_c,_day_1             3.511239e-17            1.134613e-14
pbmc:hepatitis_c,_day_2             3.429334e-12            5.584289e-10
pbmc:hepatitis_c,_day_28            1.000000e+00            5.993812e-01
pbmc:hepatitis_c,_day_7             5.993812e-01            1.000000e+00
```

```
pbmc:hepatitis_c,_day_14             2.865440e-01             6.447593e-01
pbmc:hepatitis_c,_day_0              1.649254e-16             3.164993e-16
                          pbmc:hepatitis_c,_day_14 pbmc:hepatitis_c,_day_0
pbmc:hepatitis_c,_day_1              2.127205e-15             3.169190e-41
pbmc:hepatitis_c,_day_2              2.618657e-10             2.123144e-37
pbmc:hepatitis_c,_day_28             2.865440e-01             1.649254e-16
pbmc:hepatitis_c,_day_7              6.447593e-01             3.164993e-16
pbmc:hepatitis_c,_day_14             1.000000e+00             2.643214e-20
pbmc:hepatitis_c,_day_0              2.643214e-20             1.000000e+00
```

From the output of t-statistics of both TF expression and TG activity we can see that there are three groups forming: healthy PBMCs, infected PBMCs from day 7,14,28 and infected PBMCs from day 1,2. So we can conclude that there is a clear increase in STAT1 functional activity among more recently infected PBMCs. See Wu(2012) for more details on this STAT1 analysis.

Although not displayed here, two heatmaps for the t-statistics and pvalues will also be plotted along with the data used to construct the plots. Users should always save the results, which can be done by specifying a output directory. We did not do so in this example in order to demonstrate the output of GSCAeda.

As mentioned earlier, if in the initial GSCA analysis, the argument, `directory`, is not null, then follow-up GSCA analyses will be performed and stored in the directory. This means that a similar analysis done here for GSE7123 contexts for STAT1 will be performed for each experiment ID in the GSCA results table for the TF of interest (i.e. for each of the experiments that contained contexts predicted to be enriched with significant TF functional activity). `tabSearch` and `GSCAeda` will be used recursively to search for and analyze all contexts in each experiment. All data will be stored in `directory`, which should be a filepath to a folder. Note that this process may generate a lot of data files and plots and could take some time.

References

[1] Ji, Z., Vokes, S. A., Dang, C. V., and Ji, H. (2015) Turning publicly available gene expression data into discoveries using gene set context analysis *Nucl. Acids Res.*, gkv873.

[2] Barrett T., Troup D.B., Whilhite S.E., et al. (2007) NCBI GEO: mining tens of millions of expression profiles – database and tools update. *Nucl. Acids Res.*, **35**, D760–D765.

[3] George Wu, et al. (2012) ChIP-PED enhances the analysis of ChIP-seq and ChIP-chip data. *Bioinformatics.*, **29(9)**, 1182-1189.