# Package 'ChIPXpress'

October 29, 2026

**Type** Package

**Title** ChIPXpress: enhanced transcription factor target gene identification from ChIP-seq and ChIP-chip data using publicly available gene expression profiles

**Version** 1.57.2

**Date** 2012-12-09

**Author** George Wu

**Maintainer** George Wu <georgetwu@gmail.com>

**Description** ChIPXpress takes as input predicted TF bound genes from ChIPx data and uses a corresponding database of gene expression profiles downloaded from NCBI GEO to rank the TF bound targets in order of which gene is most likely to be functional TF target.

**License** GPL(>=2)

**LazyLoad** yes

**Imports** Biobase, GEOquery, frma, affy, bigmemory, biganalytics

**Depends** R (>= 2.10), ChIPXpressData

**Suggests** mouse4302frmavecs, mouse4302.db, mouse4302cdf, RUnit, BiocGenerics

**biocViews** ChIPchip, ChIPSeq

**Config/Bioconductor/UnsupportedPlatforms** windows

**PackageStatus** Deprecated

**git_url** https://git.bioconductor.org/packages/ChIPXpress

**git_branch** devel

**git_last_commit** 674af9f

**git_last_commit_date** 2026-09-14

**Repository** Bioconductor 3.24

**Date/Publication** 2026-10-29

## Contents

ChIPXpress-package . . . . . 2
buildDatabase . . . . . 3

ChIPXpress . . . . . 5
cleanDatabase . . . . . 7
Oct4ESC_ChIPgenes . . . . . 9

**Index** **12**

---

`ChIPXpress-package` *ChIPXpress ranks TF target genes given predicted TF targets from ChIP-chip and ChIP-seq data using publicly available gene expression data*

---

## Description

Takes as input predicted TF bound targets from ChIP-chip or ChIP-seq data and ranks them according to the most likely to be a TF target gene. ChIPXpress rankings are more consistent, robust, and accurate than standard ChIP-chip or ChIP-seq rankings, which prioritize genes only on the strength of the observed peaks. ChIPXpress is able to accomplish this because it searches for TF bound genes that are highly correlated in expression with the TF across a database of highly diverse gene expresison profiles collected from different diseases, tissues, and cell types.

## Details

| | |
|---|---|
| Package: | ChIPXpress |
| Type: | Package |
| Version: | 1.22.0 |
| Date: | 2017-12-09 |
| License: | GPL-2 |

## Author(s)

Author: George Wu Maintainer: George Wu <georgetwu@gmail.com>

## References

Wu G. and Ji H. (2012) ChIPXpress: enhanced ChIP-seq and ChIP-chip target gene identification using publicly available gene expression data. *In preparation*.

buildDatabase *Builds a new database of gene expression profiles from a specified platform or sample files in NCBI GEO.*

## Description

Takes as input a specified platform (GPL ID) or vector of sample files (GSM IDs). All files for the given platform or the vector of given GSM files are then downloaded, processed by fRMA, and stored in a large matrix of expression values in big.matrix format.

The user will still need to convert the rowIDs from probeIDs to Entrez GeneIDs and then run 'cleanDatabase' on the resulting matrix in order to modify the databse to the format required for ChIPXpress analyses.

## Usage

```
buildDatabase(GPL_id, GSMfiles = NULL, SaveDir = NULL, LoadPrevious=FALSE)
```

## Arguments

| | |
|---|---|
| GPL_id | A character value of format 'GPLXXX' indicating the platform from which a database of gene expression profiles will be built. |
| GSMfiles | A vector of character values of format 'GSMXXX' indicating the samples from which a database of gene expression profiles will be built. The GSMfiles must be from the same platform, and if the GSMfiles are provided, GPL_id does not need to be specified. |
| SaveDir | Path of an EMPTY temporary directory for the built database and cel files to be downloaded to. The user will be required to create the directory beforehand and make sure it is empty. All cel files will be removed after being processed. |
| LoadPrevious | If LoadPrevious is set to TRUE, then buildDatabase will continue from the last unprocessed sample file. Use this when the buildDatabase is abruptly stopped and unable to finish. Be sure to input the other arguements exactly as they were inputted previously. |

## Details

Use this function only if you would like to build your own database of gene expression profiles. For the common user interested in ranking TF bound genes from mouse or human ChIPx data, it is easier and faster to load a pre-built database. To do, see the example code in the man page of the ChIPXpress function.

The overall process of creating a database proceeds in three steps: (1) Run buildDatabase (2) Convert rownames from probeIDs into EntrezGeneIDs (3) Run cleanDatabase The user will be required step 2 themselves. This means annotating the rows by downloading and installing the appropriate annotation package and replacing the probeID rownames with the appropriate gene IDs. Feel free to do this with any package of method that is most convenient. Note, it is possible to annotate the rows using a different ID format than the recommended Entrez GeneID. If you choose to do this, then be sure later, that when running the ChIPXpress analyses that the inputted list of TF bound genes is also of the SAME ID format.

buildDatabase uses GEOquery to download the files and then processes them using frma. Thus, the user will need to have the required frmavecs package installed apriori, so frma can process the raw gene expression files. See the help files from the frma package for more information.

The process of downloading and processing a large collection of gene expression profiles can take an extremely long time. For example, if we wanted to build a database from all of the GPL1261 samples in NCBI GEO, this will take 2 weeks since this would require processing 29086 samples! Be sure to specify an empty directory for the files to be downloaded to, since the function will remove all .gz files after each iteration in order to save hard drive space.

If for any reason the database building process is abruptly stopped, the user can continue from their previous point by running the function with the SAME inputs, save directory, and setting LoadPrevious=TRUE.

## Value

A big.matrix of gene expression values with rows corresponding to the probes on the platform and columns corresponding to the number of samples processed. This is essentially the output from frma() concatenated by column and then formatted as a big.matrix. The database will still need to be annotated by converting the probeIDs into Entrez GeneID format, and then inputted into the cleanDatabase function to obtain a finished database suitable for input into the ChIPXpress function.

## Author(s)

George Wu

## References

McCall M.N., Bolstad B.M., and Irizarry R.A. (2010) Frozen robust multiarray analysis (fRMA). Biostatistics 11, 242-253.

Barrett T., et al. (2007) NCBI GEO: mining tens of millions of expression profiles - database and tools update. Nucl. Acids Res. 35, D760-D765.

Wu G. and Ji H. (2012) ChIPXpress: enhanced ChIP-seq and ChIP-chip target gene identification using publicly available gene expression data. *In preparation*.

## Examples

```
## Not run:
## In this example, we construct a database from
## samples stored in NCBI GEO on the GPL1261 platform.
## The output would then be used as the database
## to input into the ChIPXpress function.

## Load required libraries
library(bigmemory)
library(biganalytics)
library(GEOquery)
library(affy)
library(frma)

library(mouse4302frmavecs)
## GPL1261 corresponds to the mouse430 2.0 array (required by frma)
## If your database is from a different platform, you will
## need to download the corresponding frmavecs package.

SaveDir <- tempdir()
## Make sure the save directory is empty and created.

DB <- buildDatabase(GPL_id='GPL1261',SaveDir=SaveDir)
```

```
## This will take up to 2 weeks to finish since
## it processes all GPL1261 samples currently
## stored in NCBI GEO (over 29000 samples).

## Alternatively, the user can also specify by GSM ids.
GSM_ids <- c("GSM24056","GSM24058","GSM24060","GSM24061",
             "GSM94856","GSM94857","GSM94858","GSM94859")
DB <- buildDatabase(GSMfiles=GSM_ids,SaveDir=SaveDir)
## This will only take approximately 5 minutes
## since only 8 GPL1261 samples are specified.
## Note, the database will need to include at least
## 2 samples in which the TF is above the mean TF expression
## to calculate a correlation estimate as required by the
## ChIPXpress algorithm. The more samples the better!!

## Annotates database by converting rowIDs from probeIDs into
## Entrez GeneIDs. Feel free to use any annotation method
## that is most convenient.
library(mouse4302.db)
EntrezID <- mget(as.character(rownames(DB)),mouse4302ENTREZID)
rownames(DB) <- as.character(EntrezID)

## Clean Database - cleans database into format required by ChIPXpress
cleanDB <- cleanDatabase(DB,SaveFile="newDB_GPL1261.bigmemory",
                         SavePath=SaveDir)

head(cleanDB) ## Final database ready for direct input into ChIPXpress

## To load the saved database in new R sessions, simply attach using
## the saved description file of the big.matrix as follows:
cleanDB <- attach.big.matrix("newDB_GPL1261.bigmemory.desc",path=SaveDir)
head(cleanDB)


## End(Not run)
```

---

ChIPXpress *ChIPXpress Ranking*

---

## Description

Ranks list of predicted TF bound genes from ChIPx data using a database of publicly available gene expression profiles.

## Usage

```
ChIPXpress(TFID, ChIP, DB, w=0.1, c=0, warn=FALSE, DBmu=NULL, DBvar=NULL)
```

## Arguments

| | |
|---|---|
| TFID | A character value corresponding to the Entrez GeneID of the TF-of-interest. |

ChIP — A ordered vector of Entrez GeneIDs corresponding to the predicted TF bound genes from ChIPx data. Entrez GeneIDs should be reported as character values and sorted from the most likely to the least likely to be bound by the TF based on the strength of peak signals near each gene in ChIPx data.

DB — A big.matrix of expression values that is used as the database of publicly availiable gene expression profiles. This database can be either built by the user directly using the functions 'build database' and 'clean database', or loaded from a pre-built database in package *ChIPXpressData*. *ChIPXpressData* currently contains two pre-built databases avaliable for analysis purposes - GPL1261 for mouse ChIPx data and GPL570 for human ChIPx data. See example code below on how to load a big.matrix formatted gene expression database.

w — A numeric value specifying the weight used to combine the ChIPx and gene expression compendium based rankings. The typical user will not need to modify the default value. Specifically, if we let Pg be the rank of gene g based on ChIPx data and Ag be the rank of gene g based on gene expression compendium data, the final ChIPXpress score for gene g is w*Pg+(1-w)*Ag. ChIPXpress rankings are then sorted from the smallest to the largest baed on the ChIPXpress scores. Thus, for $w < 0.5$, the gene expression information has a larger impact on the final ChIPXpress rankings, and for $w > 0.5$, the input ChIPx rankings has a larger inpact on the final ChIPXpress rankings. Tests using real datasets revealed that w=0.1 is the optimal weight.

c — A numeric value specifying the TF expression cutoff. The typical user will not need to modify the default value. The TF expression cutoff, c, is used to remove samples in which the TF expression is below the TF expression cutoff, c, prior to the calculation of the absolute correlation. Testing using real datasets revealed that c=0 is the optimal TF expression cutoff.

warn — If set to TRUE, then will report warning if mean, variance, and coefficient of variation of the TF expression.

DBmu — Mean of each probe in the database prior to standardization. Can be found for the pre-built databases in the ChIPXpressData package.

DBvar — Variance of each probe in the database prior to standardization. Can be found for the pre-built databases in the ChIPXpressData package.

## Details

ChIPXpress works by ranking the TF bound genes predicted from ChIPx data by the absolute correlation between each gene and the TF in the database of gene expression profiles. Genes that are more highly correlated, either positively or negatively, are ranked as the most likely to be an actual TF target gene. Note, the absolute correlation is calculated only using the samples in which the TF expression is above c=0, i.e. the mean TF expression across all samples, in order to improve prediction performance. See reference below for more information on the rationale behind the ranking algorithm of ChIPXpress and the selection of the default values w=0.1 and c=0.

## Value

Returns a list with two vectors: the first vector contains the absolute correlations of each predicted TF bound gene with the TF in ranked order, where the names of the vector correspond to the Entrez GeneID of each gene, and the second vector contains the Entrez GeneIDs of the predicted TF bound genes not found in the database. An additional warning will be given if DBvar is specified, which will let the user know if the TF has a low expression variance in the database.

### Author(s)

George Wu

### References

Wu G. and Ji H. (2012) ChIPXpress: enhanced ChIP-seq and ChIP-chip target gene identification using publicly available gene expression data. *In preparation*.

### Examples

```
## Example analyses of real Oct4 bound genes predicted
## from ChIP-seq data in ESC using pre-built GPL1261
## database

## Load predicted Oct4-bound genes from ChIPx data
data(Oct4ESC_ChIPgenes)

## Load example GPL1261 Database
library(bigmemory)
path <- system.file("extdata",package="ChIPXpressData")
DB_GPL1261 <- attach.big.matrix("DB_GPL1261.bigmemory.desc",path=path)
## To load the human GPL570 data, replace 'DB_GPL1261' with 'DB_GPL570'.

## Run ChIPXpress ("18999" is Entrez GeneID of Oct4)
out <- ChIPXpress(TFID="18999",ChIP=Oct4ESC_ChIPgenes$EntrezID,DB=DB_GPL1261)
head(out[[1]]) ## ChIPXpress Rankings
head(out[[2]]) ## Missing genes not found

## For the final step, you can convert the Output into a
## clean table with genes names or any other preferred gene identifier
## by using any of your favorite annotation packages (e.g., biomaRt).
## Here, we can use the original Oct4ESC_ChIPgenes dataframe to do so directly.
GeneNames <- Oct4ESC_ChIPgenes$Annotation[match(names(out[[1]]),Oct4ESC_ChIPgenes$EntrezID)]
Result <- data.frame(1:length(out[[1]]),GeneNames,names(out[[1]]),out[[1]])
colnames(Result) <- c("Rank","GeneNames","EntrezID","ChIPXpressScore")
head(Result) ## Clean ChIPXpress rankings
```

---

## cleanDatabase *Cleans the annotated output from buildDatabase to the required format for ChIPXpress analyses*

---

### Description

Given the output matrix of 'buildDatabase' after the rows have been properly annotated (preferably into Entrez GeneIDs), this function will clean the database such that each row is normalized and in 1-to-1 correspondance with a single Entrez GeneID or alternative gene ID format.

### Usage

```
cleanDatabase(DB, SaveFile="newDB.bigmemory", SavePath=".")
```

## Arguments

| | |
|---|---|
| DB | A big.matrix of expression values from 'buildDatabase' after the rows of the matrix are converted into Entrez GeneID format. Alternative gene ID formats for the row names is also suitable. |
| SaveFile | Specifies the name of the big.matrix and big.matrix description file. The default file name is "newdb.bigmemory". This will be used to load the database for future R sessions. |
| SavePath | Specifies the directory in which the final big.matrix and big.matrix descrption file is saved. The default is to save into the current working directory. |

## Details

This function is to be used after buildDatabase and row annotation has already been completed. Specifically, the entire process of creating a database proceeds in the following three steps: (1) Run buildDatabase (2) Convert rownames from probeIDs into EntrezGeneIDs (3) Run cleanDatabase The user will be required to complete step 2 themselves. This means annotating the rows by downloading and installing the appropriate annotation package and replacing the probeID rownames with the appropriate gene IDs. Feel free to do this with any package or method that is most convenient. Note, it is possible to annotate the rows using a different ID format than the recommended Entrez GeneID. If you choose to do this, then be sure later, that when running the ChIPXpress analyses that the inputted list of TF bound genes is also of the SAME ID format.

cleanDatabase will first find all rownames that corresponds to multiple rows and then retain only the row with expression values with the highest variance. This is to ensure each row corresponds to a single gene ID (or Entrez GeneID). Then the function will normalize by rows - subtracting by the mean and dividing by the standard deviation.

## Value

A big.matrix of normalized gene expression values. Each row will uniquely annotated to a single Entrez GeneID or alternative gene ID format. The big.matrix is ready for input as the database of gene expression profiles utilized by the ChIPXpress function.

## Author(s)

George Wu

## References

McCall M.N., Bolstad B.M., and Irizarry R.A. (2010) Frozen robust multiarray analysis (fRMA). Biostatistics 11, 242-253.

Barrett T., et al. (2007) NCBI GEO: mining tens of millions of expression profiles - database and tools update. Nucl. Acids Res. 35, D760-D765.

Wu G. and Ji H. (2012) ChIPXpress: enhanced ChIP-seq and ChIP-chip target gene identification using publicly available gene expression data. *In preparation*.

## Examples

```
## Not run:
## In this example, we construct a database from
## samples stored in NCBI GEO on the GPL1261 platform.
## The output would then be used as the database
## to input into the ChIPXpress function.
```

```
## Load required libraries
library(bigmemory)
library(biganalytics)
library(GEOquery)
library(affy)
library(frma)

library(mouse4302frmavecs)
## GPL1261 corresponds to the mouse430 2.0 array (required by frma)
## If your database is from a different platform, you will
## need to download the corresponding frmavecs package.

SaveDir <- tempdir()
## Make sure the save directory is empty and created.

DB <- buildDatabase(GPL_id='GPL1261',SaveDir=SaveDir)
## This will take up to 2 weeks to finish since
## it processes all GPL1261 samples currently
## stored in NCBI GEO (over 29000 samples).

## Alternatively, the user can also specify by GSM ids.
GSM_ids <- c("GSM24056","GSM24058","GSM24060","GSM24061",
             "GSM94856","GSM94857","GSM94858","GSM94859")
DB <- buildDatabase(GSMfiles=GSM_ids,SaveDir=SaveDir)
## This will only take approximately 5 minutes
## since only 8 GPL1261 samples are specified.
## Note, the database will need to include at least
## 2 samples in which the TF is above the mean TF expression
## to calculate a correlation estimate as required by the
## ChIPXpress algorithm. The more samples the better!!

## Annotates database by converting rowIDs from probeIDs into
## Entrez GeneIDs. Feel free to use any annotation method
## that is most convenient.
library(mouse4302.db)
EntrezID <- mget(as.character(rownames(DB)),mouse4302ENTREZID)
rownames(DB) <- as.character(EntrezID)

## Clean Database - cleans database into format required by ChIPXpress
cleanDB <- cleanDatabase(DB,SaveFile="newDB_GPL1261.bigmemory",
                         SavePath=SaveDir)

head(cleanDB) ## Final database ready for direct input into ChIPXpress

## To load the saved database in new R sessions, simply attach using
## the saved description file of the big.matrix as follows:
cleanDB <- attach.big.matrix("newDB_GPL1261.bigmemory.desc",path=SaveDir)
head(cleanDB)


## End(Not run)
```

`Oct4ESC_ChIPgenes` *Predicted Oct4 bound genes in embryonic stem cells (ESC) obtained from analyzing ChIP-seq data*

---

## Description

Vector of Oct4 bound genes predicted by analyzing ESC ChIP-seq data from GSE11724.

## Usage

```
data(Oct4ESC_ChIPgenes)
```

## Format

A data frame with 5158 observations on the following 21 variables.

`Rank` a numeric vector

`Chr` a character vector

`Start` a numeric vector

`End` a numeric vector

`Strand` a character vector

`Annotation` a character vector

`Gene` a character vector

`EntrezID` a numeric vector

`peak_length` a numeric vector

`FDR` a numeric vector

`left_peakboundary` a numeric vector

`right_peakboundary` a numeric vector

`peak_summit` a numeric vector

`bound_center` a numeric vector

`bound_width` a numeric vector

`maxT` a numeric vector

`maxT_pos` a numeric vector

`max_log2FC` a numeric vector

`maxFC_pos` a numeric vector

`minuslog10_minPoisP` a numeric vector

`minPoisP_pos` a numeric vector

## Details

To obtain the TF bound gene predictions, the ChIP-seq data is processed using CisGenome with the default parameters. Only peaks significant at a FDR of 0.10 are retained and annotated by assigning peaks to genes if the peak falls within 10kbp upstream or 5kbp downstream of the gene transcription start site. Only the highest ranking peak for each gene is retained in the data frame for input. To be clear, the Rank in the data frame corresponds to the original peak ranking by CisGenome. The ChIPx ranking is simply the order of genes in the data frame.

## Source

www.ncbi.nlm.nih.gov/geo/

## References

Marson A. et al. (2008) Connecting microRNA genes to the core trancsriptional regulatory circuitry of embryonic stem cells. *Cell* **134**, 521-533.

Barrett T., et al. (2007) NCBI GEO: mining tens of millions of expression profiles - database and tools update. *Nucl. Acids Res.* **35**, D760-D765.

## Examples

```
data(Oct4ESC_ChIPgenes)
```

# Index

∗ **ChIPXpress**
ChIPXpress, 5
∗ **buildDatabase**
buildDatabase, 3
∗ **cleanDatabase**
cleanDatabase, 7
∗ **datasets,Oct4**
Oct4ESC_ChIPgenes, 10
∗ **package, ChIPXpress**
ChIPXpress-package, 2

buildDatabase, 3

ChIPXpress, 5
ChIPXpress-package, 2
cleanDatabase, 7

Oct4ESC_ChIPgenes, 9