

# Package ‘titan’

February 15, 2012

**Version** 1.0-16

**Date** 2009-11-20

**Title** Titration analysis for mass spectrometry data

**Author** Tom Price <t0mpr1c3@gmail.com>

**Maintainer** Tom Price <t0mpr1c3@gmail.com>

**Depends** R (>= 2.7.0), MASS, boot, tcltk, splines, lattice

**Suggests**

**Description** GUI to analyze mass spectrometric data on the relative abundance of two substances from a titration series.

**License** GPL (>= 2)

**URL** <http://sgdp.iop.kcl.ac.uk/tprice/software.html>

**Repository** CRAN

**Date/Publication** 2009-11-20 11:37:49

## R topics documented:

hypoxia	2
titan	2

**Index** 7

---

hypoxia

*Gene Expression Under Hypoxia/Normoxia*

---

**Description**

Gene expression of MCF7 cells under conditions of hypoxia and normoxia.

**Usage**

data(hypoxia)

**Format**

A data frame with 1009 observations on 5 variables.

freq cDNA frequency (numeric, 0-1)

gene Gene (factor)

rx Treatment condition (factor)

comp.conc Concentration of competitor (numeric, M)

flag Exclude data point (numeric, 0 or 1)

**Details**

Further details are available from the reference below.

**Author(s)**

Tom Price <t0mpr1c3@gmail.com>

**References**

Elvidge, G., Price, T. S., Glenny, L. and Ragoussis, I. *Development and evaluation of real-competitive PCR for high-throughput quantitative applications*. *Anal Biochem*. 2005;339(2):231-241.

---

titan

*Titration Analysis*

---

**Description**

GUI to analyze mass spectrometric data on the relative abundance of two substances from a titration series.

**Usage**

```
titan(
  data =      NULL,
  trace =     TRUE,
  widget =    TRUE,
  dataFile =  "",
  outFile =   "",
  pdfFile =   "",
  flagRaw =   FALSE,
  flagFitted = FALSE,
  freqLo =    .05,
  freqHi =    .95,
  reg =       "least.squares",
  term =      "linear",
  sel =       "wald",
  alpha =     .05,
  rx0 =       NULL,
  rx1 =       NULL,
  gene0 =     NULL,
  gene1 =     NULL,
  R =         1000,
  seed =      0,
  ciConf =    .95,
  ciType =    "all"
)
```

**Arguments**

<code>data</code>	A data frame containing the factors <code>gene</code> and <code>treatment</code> (or <code>rx</code> ), the numeric vectors <code>frequency</code> and <code>competitor</code> (or <code>concentration</code> ), and the numeric vector <code>flag</code> taking binary values. <code>gene</code> and <code>treatment</code> specify the sample, and <code>competitor</code> the concentration of the competitor. <code>flag</code> takes the value 0 for data points that are to be used in the analysis, and 1 for data points that are to be ignored. If this parameter is set to <code>NULL</code> , the user must supply the address of a data file containing this information: see <code>dataFile</code> below.
<code>trace</code>	A logical scalar. If <code>TRUE</code> , the output is printed as the function proceeds.
<code>widget</code>	A logical scalar. If <code>TRUE</code> , the GUI is used to specify the data and set parameters for the analysis. Otherwise, the following arguments must be supplied:
<code>dataFile</code>	A character string specifying the location of the input data file. The file should be a tab-delimited text file with column headings containing the fields specified for the <code>data</code> parameter above.
<code>pdfFile</code>	A character string specifying the location in which to save the output pdf file. If set to a null string, the output is printed on the default device.
<code>outFile</code>	A character string specifying the location in which to save the output text file. If set to a null string, the output is printed on the default device.
<code>freqLo</code>	A numeric value between 0 and 1 specifying the lower bound for cDNA frequency.

freqHi	A numeric value between 0 and 1 specifying the higher bound for cDNA frequency.
flagRaw	A logical value. If TRUE, flag raw cDNA frequency values outside the range specified by freqLo and freqHi.
flagFitted	A logical value. If TRUE, flag fitted cDNA frequency values outside the range specified by freqLo and freqHi.
reg	A character string specifying the type of regression to be used: "least.squares", "robust", or "spline" for least squares, robust, or natural spline regressions, respectively.
term	A character string specifying the terms to be entered into the regression. Least squares and robust regressions may take the values "linear", "quadratic" or "parallel.linear". "linear" fits a regression with a linear term and intercept for each combination of gene and treatment condition. "quadratic" also fits quadratic terms. "parallel.linear" estimates only one linear term per gene, with intercepts for each combination of gene and treatment. Natural spline regressions specify the number of degrees of freedom in the regression as either a numeric or character vector.
sel	A character string identifying the selection method for terms in the regression. For least squares regressions, the two options are "wald", indicating backwards selection by the Wald test, or "aic", signifying stepwise selection by AIC. Robust regressions use only backwards selection by the Wald test. Natural spline regressions use only stepwise selection by AIC.
alpha	A numeric scalar containing the value of alpha (maximum p-value) for selection by the Wald test.
rx0	A character string specifying the name of the baseline level of the factor treatment, against which other treatments are compared when calculating fold changes. If NULL, the first level of treatment is taken as the baseline condition.
rx1	A character vector containing the names of the levels of the factor treatment that are entered into the analysis as test conditions when calculating fold changes. If NULL, it is assumed that all the treatments other than that specified in rx0 are test conditions.
gene0	A character vector containing the names of the 'housekeeping' or control genes in the factor gene. If NULL, it is assumed that none of the genes are housekeeping genes.
gene1	A character vector containing the names of the test genes in the factor gene. If NULL, it is assumed that all the genes other than those specified in gene0 are test genes.
R	A numeric scalar specifying the number of bootstrap replicates to be performed. If zero, no bootstrap analysis is performed.
seed	A numeric scalar containing the seed for the random number generator used in the bootstrap. If NULL, a random seed is supplied.
ciConf	A scalar or vector containing the confidence level(s) of the required bootstrap interval(s).
ciType	A vector of character strings representing the type of intervals required. The value should be any subset of the values c("norm", "basic", "perc", "bca") or simply "all" which will compute all four types of intervals.

## Details

Further details are available from the references below.

## Value

An object of class "titan" containing the output. it has components

data	A data frame containing the data as input to the function, including flag which contains the actual data flags used in the analysis.
opt	A list of arguments supplied to the function, either directly or using the GUI.
reg	An object of class lm or rlm, the output from the regression.
interpolation	A list containing a matrix roots containing the interpolated values for each combination of gene and treatment, i.e. the equivalence points. If there is more than one treatment, the list contains a further matrix log10fold containing the base 10 logarithms of the fold changes, calculated relative to the baseline treatment, and adjusted for the fold changes for the housekeeping or control genes.
boot	An object of class boot, the output from the bootstrap.
bootci	An object of class titanbootci containing the bootstrap confidence intervals.

## Author(s)

Tom Price <t0mpr1c3@gmail.com>

## References

Elvidge, G., Price, T. S., Glenny, L. and Ragoussis, I. *Development and evaluation of real-competitive PCR for high-throughput quantitative applications*. Anal Biochem. 2005;339(2):231-241.

<http://sgdp.iop.kcl.ac.uk/tprice/titan/userguide.pdf>

## Examples

```
data(hypoxia)

# Analyze titration series using GUI (code not run):
#
# t1 <- titan(data = hypoxia, R=0)

# Perform an identical reanalysis without using GUI:
#
# first get previous options
#
opt <- list(data = hypoxia, R=0)
#
# adjust the options so that
# no further data points are flagged
#
opt$flagRaw <- FALSE
opt$flagFitted <- FALSE
#
```

```
# turn off GUI
#
opt$widget <- FALSE
#
# call function
t2 <- do.call( "titan", opt )

# Reanalyze data with different parameters:
#
opt$reg <- "robust"
opt$sel <- "wald"
t3 <- do.call( "titan", opt )
```

# Index

\*Topic **misc**  
hypoxia, 2  
titan, 2

hypoxia, 2

titan, 2