Package 'ImpulseDE'

January 23, 2020

Type Package

Title Detection of DE genes in time series data using impulse models

Version 1.13.0

Date 2016-06-16

Author Jil Sander [aut, cre], Nir Yosef [aut]

Maintainer Jil Sander <jil.sander@uni-bonn.de>, Nir Yosef <niryosef@berkeley.edu>

biocViews Software, StatisticalMethod, TimeCourse

Depends graphics, grDevices, stats, utils, parallel, compiler, R (>= 3.2.3)

Imports amap, boot

Suggests longitudinal, knitr

Description ImpulseDE is suited to capture single impulse-like patterns in high throughput time series datasets. By fitting a representative impulse model to each gene, it reports differentially expressed genes whether across time points in a single experiment or between two time courses from two experiments. To optimize the running time, the code makes use of clustering steps and multi-threading.

License GPL-3

LazyData TRUE

RoxygenNote 5.0.0

VignetteBuilder knitr

URL https://github.com/YosefLab/ImpulseDE

BugReports https://github.com/YosefLab/ImpulseDE/issues

NeedsCompilation no

git_url https://git.bioconductor.org/packages/ImpulseDE

git_branch master

git_last_commit a64fac7

git_last_commit_date 2019-10-29

Date/Publication 2020-01-22

R topics documented:

`calc_impulse` *Impulse model value prediction*

Description

Calculates impulse model values for given timepoints and predicted impulse parameters.

Usage

```
calc_impulse(theta, timepoints)
```

Arguments

`theta`	numerical vector of impulse parameters with the order beta, h0, h1, h2, t1, t2.
`timepoints`	numercial vector of time point(s).

Value

The predicted impulse model values for the given time point(s).

Author(s)

Jil Sander

References

Chechik, G. and Koller, D. (2009) Timing of Gene Expression Responses to Envi-ronmental Changes. J. Comput. Biol., 16, 279-290.

See Also

`impulse_DE`, `plot_impulse`.

Examples

```
#' theta vector in the order beta, h0, h1, h2, t1, t2
theta <- c(9.9, 14.7, 17.0, 16.9, -0.1, 37.0)
#' time points
timepoints <- c(0, 2, 4, 6, 8, 18, 24, 32, 48, 72)
#' calculate impulse values
impulse_values <- calc_impulse(theta, timepoints)
```

impulse_DE *Differential expression analysis using impulse models*

Description

Fits an impulse model to time course data and uses this model as a basis to detect differentially expressed (DE) genes. If a single time course data set is given, DE genes are detected over time, whereas if an additional control time course data set is present, DE genes are detected between both datasets.

Usage

```
impulse_DE(expression_table = NULL, annotation_table = NULL,
  colname_time = NULL, colname_condition = NULL,
  control_timecourse = FALSE, control_name = NULL, case_name = NULL,
  expr_type = "Array", plot_clusters = TRUE, n_iter = 100,
  n_randoms = 50000, n_process = 4, Q_value = 0.01, new_device = TRUE)
```

Arguments

expression_table	numeric matrix of expression values; genes should be in rows, samples in columns. Data should be properly normalized and log2-transformed as well as filtered for present or variable genes.
annotation_table	table providing co-variables for the samples including condition and time points. Time points must be numeric numbers.
colname_time	character string specifying the column name of the co-variable "Time" in annotation_table
colname_condition	character string specifying the column name of the co-variable "Condition" in annotation_table
control_timecourse	logical indicating whether a control time timecourse is part of the data set (TRUE) or not (FALSE). Default is FALSE.
control_name	character string specifying the name of the control condition in annotation_table.
case_name	character string specifying the name of the case condition in annotation_table. Should be set if more than two conditions are present in annotation_table.
expr_type	character string with allowed values "Array" or "Seq". Default is "Array".
plot_clusters	logical indicating whether to plot the clusters (TRUE) or not (FALSE). Default is TRUE.
n_iter	numeric value specifying the number of iterations, which are performed to fit the impulse model to the clusters. Default is 100.
n_randoms	numeric value specifying the number of generated randomized background iterations, which are used for differential expression analysis. Default is 50000 and this value should not be decreased.
n_process	numeric value indicating the number of processes, which can be used on the machine to run calculations in parallel. Default is 4. The specified value is internally changed to min(detectCores() -1,n_process) using the detectCores function from the package parallel to avoid overload.

Q_value	numeric value specifying the cutoff to call genes significantly differentially expressed after FDR correction (adjusted p-value). Default is 0.01.
new_device	logical indicating whether each plot should be plotted into a new device (TRUE) or not (FALSE). Default is TRUE.

Details

ImpulseDE is based on the impulse model proposed by Chechik and Koller, which reflects a two-step behavior of genes within a cell responding to environmental changes (Chechik and Koller, 2009). To detect differentially expressed genes, a five-step workflow is followed:

1. The genes are clustered into a limited number of groups using k-means clustering. If plot_clusters = TRUE, the clusters are plotted.
2. The impulse model is fitted to the mean expression profiles of the clusters. The best parameter sets are then used for the next step.
3. The impulse model is fitted to each gene separately using the parameter sets from step 2 as optimal start point guesses.
4. The impulse model is fitted to a randomized dataset (bootstrap), which is essential to detect significantly differentially expressed genes (Storey et al., 2005).
5. Detection of differentially expressed genes utilizing the fits to the real and randomized data sets. FDR-correction is performed to obtain adjusted p-values (Benjamini and Hochberg, 1995).

Value

List containing the following elements:

- impulse_fit_results List containing fitted values and model parameters:
  - impulse_parameters_case Matrix of fitted impulse model parameters and sum of squared fitting errors for the case dataset. If a control time course is present, corresponding list entries will exist for the control and the combined dataset as well (named impulse_parameters_control and impulse_parameters_combined, respectively).
  - impulse_fits_case Matrix of impulse values calculated based on the analyzed time points and the fitted model parameters for the combined dataset. If a control time course is present, corresponding list entries will exist for the control and the combined dataset as well (named impulse_fits_control and impulse_fits_combined, respectively).
- DE_results List containg the results from the differential expression analysis:
  - DE_genes data.frame containing the names of genes being called as differentially expressed according to the specified cutoff Q_value together with the adjusted p-values.
  - pvals_and_flags data.frame containing all gene names together with the adjusted p-values and flags for differential expression according to additional tests.
- clustering_results List containing the clustering results:
  - kmeans_clus_case Numeric vector of clusters IDs, to which the genes were finally assigned.
  - cluster_means_case Matrix containing the mean expression values for each cluster (taken over all genes assigned to a cluster).
  - pre_clus_case Numeric number of clusters determined after the first (preliminary) clustering step.
  - fine_clus_case Numeric number of final clusters determined after the second clustering step.

If a control time course is present, those four list entries will exist correspondingly for the control and the combined dataset as well (ending with `_control` and `_combined` instead of `_case`, respectively).

Author(s)

Jil Sander

References

Benjamini, Y. and Hochberg, Y. (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. J. R. Stat. Soc. Series B Stat. Methodol., 57, 289-300.

Storey, J.D. et al. (2005) Significance analysis of time course microarray experiments. Proc. Natl. Acad. Sci. USA, 102, 12837-12841.

Rangel, C., Angus, J., Ghahramani, Z., Lioumi, M., Sotheran, E., Gaiba, A., Wild, D.L., Falciani, F. (2004) Modeling T-cell activation using gene expression profiling and state-space models. Bioinformatics, 20(9), 1361-72.

Chechik, G. and Koller, D. (2009) Timing of Gene Expression Responses to Envi-ronmental Changes. J. Comput. Biol., 16, 279-290.

Yosef, N. et al. (2013) Dynamic regulatory network controlling TH17 cell differentiation. Nature, 496, 461-468.

See Also

`plot_impulse`, `calc_impulse`.

Examples

```
#' Install package longitudinal and load it
library(longitudinal)
#' Attach datasets
data(tcell)
#' check dimension of data matrix of interest
dim(tcell.10)
#' generate a proper annotation table
annot <- as.data.frame(cbind("Time" =
   sort(rep(get.time.repeats(tcell.10)$time,10)),
   "Condition" = "activated"), stringsAsFactors = FALSE)
#' Time columns must be numeric
annot$Time <- as.numeric(annot$Time)
#' rownames of annotation table must appear in data table
rownames(annot) = rownames(tcell.10)
#' apply ImpulseDE in single time course mode
#' since genes must be in rows, transpose data matrix using t()
#' For the example, reduce iterations to 10, randomizations to 50, number of
#' genes to 20 and number of used processors to 1:
impulse_results <- impulse_DE(t(tcell.10)[1:20,], annot, "Time", "Condition",
   n_iter = 10, n_randoms = 50, n_process = 1)
```

plot_impulse *Plot impulse model fits*

Description

Plots impulse model fits for the specified gene IDs. In the case of two time courses, the fits for the combined, case and control data are plotted.

Usage

```
plot_impulse(gene_IDs, data_table, data_annotation, imp_fit_genes,
  control_timecourse = FALSE, control_name = NULL, case_name = NULL,
  file_name_part = "", title_line = "", sub_line = "",
  new_device = TRUE)
```

Arguments

gene_IDs	character vector of gene names to be plotted; must be part of the rownames of data_table
data_table	numeric matrix of expression values; genes should be in rows, samples in columns. Data should be properly normalized and log2-transformed as well as filtered for present or variable genes.
data_annotation	table providing co-variables for the samples including condition and time points. Time points must be numeric numbers.
imp_fit_genes	list of fitted impulse model values and parameters as produced by impulse_DE (list element impulse_fit_results).
control_timecourse	logical indicating whether a control time timecourse is part of the data set (TRUE) or not (FALSE). Default is FALSE.
control_name	character string specifying the name of the control condition in annotation_table.
case_name	character string specifying the name of the case condition in annotation_table. Should be set if more than two conditions are present in data_annotation.
file_name_part	character string to be used as file extention.
title_line	character string to be used as title for each plot.
sub_line	character string to be used as subtitle for each plot.
new_device	logical indicating whether each plot should be plotted into a new device (TRUE) or not (FALSE). Default is TRUE.

Value

Plots of the impulse model fits for the specified gene IDs.

Author(s)

Jil Sander

References

Chechik, G. and Koller, D. (2009) Timing of Gene Expression Responses to Envi-ronmental Changes. J. Comput. Biol., 16, 279-290.

See Also

impulse_DE, calc_impulse.

Examples

```
#' Install package longitudinal and load it
library(longitudinal)
#' Attach datasets
data(tcell)
#' check dimension of data matrix of interest
dim(tcell.10)
#' generate a proper annotation table
annot <- as.data.frame(cbind("Time" =
     sort(rep(get.time.repeats(tcell.10)$time,10)),
     "Condition" = "activated"), stringsAsFactors = FALSE)
#' Time columns must be numeric
annot$Time <- as.numeric(annot$Time)
#' rownames of annotation table must appear in data table
rownames(annot) = rownames(tcell.10)
#' since genes must be in rows, transpose data matrix using t()
#' consider 6 genes for now only
genes <- c("SIVA","CD69","ZNFN1A1","IL4R","MAP2K4","JUND")
tcell.10.filtered <- t(tcell.10[,genes])
#' generate a list object having the form of the output of impulse_DE
#' first the parameter fits and SSEs
impulse_parameters_case <- matrix(
    c(0.6,18.6,17.2,17.4,5.1,40.2,3.5,
      0.3,-464.9,18.3,17.3,-17.2,35.3,17.5,
      23.2,18,18.8,18.5,3,37,13.2,
      NA,NA,NA,NA,NA,NA,3.1,
      NA,NA,NA,NA,NA,NA,9.6,
      9.5,17.5,18.7,17.5,8,48,46.7),length(genes),7, byrow = TRUE)
rownames(impulse_parameters_case) <- genes
colnames(impulse_parameters_case) <- c("beta", "h0", "h1", "h2", "t1", "t2", "SSE")
#' then the fitted values for the time points
impulse_fits_case <- matrix(c(
    18.55,18.43,18.15,17.73,17.43,17.24,17.24,17.24,17.38,17.38,
    16.22,17.18,17.7,17.97,18.12,18.27,18.26,18.03,17.3,17.28,
    18,18,18.82,18.82,18.82,18.82,18.82,18.82,18.48,18.48,
    15.93,15.93,15.93,15.93,15.93,15.93,15.93,15.93,15.93,15.93,
    17.62,17.62,17.62,17.62,17.62,17.62,17.62,17.62,17.62,17.62,
    17.5,17.5,17.5,17.5,18.18,18.67,18.67,18.67,17.98,17.53)
    ,length(genes),length(unique(annot$Time)), byrow = TRUE)
rownames(impulse_fits_case) <- genes
colnames(impulse_fits_case) <- unique(annot$Time)
#' finalize list object
impulse_fit_genes <- list("impulse_parameters_case" = impulse_parameters_case,
                          "impulse_fits_case" = impulse_fits_case)
#' Plot expression values
plot_impulse(genes, tcell.10.filtered, annot, impulse_fit_genes)
```

Index