ImpulseDE

Jil Sander

June 16th, 2016

Table of Contents

Introduction

ImpulseDE detects differentially expressed (DE) genes in high-throughput time course experiments. It accepts two different kinds of inputs: whether a single time course dataset (1TK) or a dataset containing a case as well as a control time course for each time point measured (2TK). For the first scenario, it identifies genes being differentially expressed across time points. For the second scenario, *ImpulseDE* reports genes being differentially expressed between both conditions. *ImpulseDE* follows a five-step workflow:

Step	Explanation
Clustering	The genes are clustered into a limited number of groups using k-means. In default modus, *ImpulseDE* prints the plots for each cluster.
Fit to clusters	*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). This model is fitted to the mean expression profiles of the clusters.
Fit to genes	The best parameter sets obtained from the clusters are then used to fit an impulse model to each gene.
Fit to random data	The impulse model is fitted to a randomized dataset (bootstrap), which is essential to detect differentially expressed genes (Storey et al. 2005).
Detection of differentially expressed genes	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).

Requirements for input data

ImpulseDE requires an expression table (only numbers) as well as an annotation table (characters allowed). The requirements for the two tables are the following:

Feature	Explanation
Expression table	Genes have to be in rows and samples in columns. Both rows and columns should have unique identifiers.
Annotation table	Must have two columns, one carrying the timestamps as numeric numbers and the other one carrying the condition information. In the case of two time courses, two conditions are required. More than two conditions are allowed to be specified, but then two conditions (one case and one control conditions) have to be specified by the user for each run separately. The samples (row names) do not have to have the same order as in the expression table (column names), but the sample identifiers must be identical. Additional columns are allowed but will be ignored later on.
Missing values	Are **not supported**. Genes having missing values for at least one sample will be excluded from the analysis.
Time points	Since the parametric model contains six parameters, the dataset should contain at least **six** time points.
Normalization and filtering	Gene expression data should be **properly normalized** including **log2-transformation** and **filtered** to avoid spending time on fitting the model to non-informative genes (e.g. not expressed or not variable genes). No impulse model will be fitted to genes having a coefficient of variation less than 0.025; instead, the mean across all samples is returned as the "fit".

Calling ImpulseDE

The first four input parameters have to be specified by the user. Those are the names of the two input tables (*expression_table* and *annotation_table*) as well as the two column names carrying the time (*colname_time*) and condition (*colname_condition*) information within the annotation table. Additional parameters can be set to specify the time course scenario as well as fitting and parallelization parameters.
In the default modus, *ImpulseDE* expects a single time course scenario without any control data (*control_timecourse = FALSE* and *control_name = NULL*). In the case of two time courses, the *control_timecourse* parameter has to be set to *TRUE* and for *control_name* the name of the control within *colname_condition* has to be specified. If more than two conditions are present within the annotation table, *case_name* has to be set in addition to run *ImpulseDE* for the desired case condition.
Regarding the fitting, as default *ImpulseDE* will run 100 iterations (*n_iter = 100*) to optimize the model parameters, generate 50.000 random data points (*n_randoms = 50.000*) to estimate bootstrapped p-values for DE analysis, and determines DE genes using an FDR-adjusted p-value cutoff (q-value) of 1% (*Q_value = 0.01*). Furthermore, in default modus it will split the run into 4 processes (*n_process = 4*). If parallelization is not possible on the device or is not admired, *n_process* should be set to *1*.

Output of ImpulseDE

ImpulseDE returns a list consisting of three sublists: *impulse_fit_results*, *DE_results* and *clustering_results*. The first contains the fitted impulse model parameters, sum of squared fitting errors as well as the calculated impulse values for all time points. The second provides the names of the genes being called as differentially expressed according to a specified cutoff together with the adjusted p-values (*DE_genes*) as well as the adjusted p-values, flags and results of additional tests for all genes (*pvals_and_flags*). The third speciFIes the clusters, to which the genes were assigned to as well as the mean expression values for the clusters.

Differential expression analysis for single time courses

In the case of a single time course experiment, *ImpulseDE* will detect differentially expressed genes over time. A fitting dataset is provided within the R package *longitudinal*, where T cells were stimulated with PMA and ionomicin and harvested at 10 different time points (Rangel et al. 2004). The dataset contains 10 measurements per time point for 58 genes:

```
# (Install package longitudinal) and load it
library(longitudinal)
```

```
## Loading required package: corpcor
```

```
# attach T cell data
data(tcell)
# check dimension of data matrix of interest
dim(tcell.10)
```

```
## [1] 100  58
```

In order to be able to apply *ImpulseDE* on this dataset, *tcell.10* has to be transposed using *t()* during the call since genes need to be in rows and samples in columns. Additionally, it is necessary to create a proper annotation table:

```
# generate annotation table with columns "Time" and "Condition"
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)
head(annot)
```

```
##     Time Condition
## 0-1    0 activated
## 0-2    0 activated
## 0-3    0 activated
## 0-4    0 activated
## 0-5    0 activated
## 0-6    0 activated
```

It is important that the *Time* column contains numeric values and that the *Condition* column is not a factor. Since the dataset contains only a sinlge time course and therefore only one condition, the *Condition* column contains only one unique value, *activated*.

Running the analysis 1TK

ImpulseDE provides a single function, *impulse_DE*, which runs all the analysis steps automatically and prints the current status on the screen. To run *ImpulseDE* with all default options, only four variables need to be set for the single time course scenario: *expression_table*, *annotation_table*, *colname_time* and *colname_condition*. However, for demonstration purposes the number of iterations, randomizations as well as the number of used processors will be reduced. For real datasets, it not recommended to reduce *n_iter* as well as *n_randoms*. Additionally, the analysis will be limited to the first 20 genes:

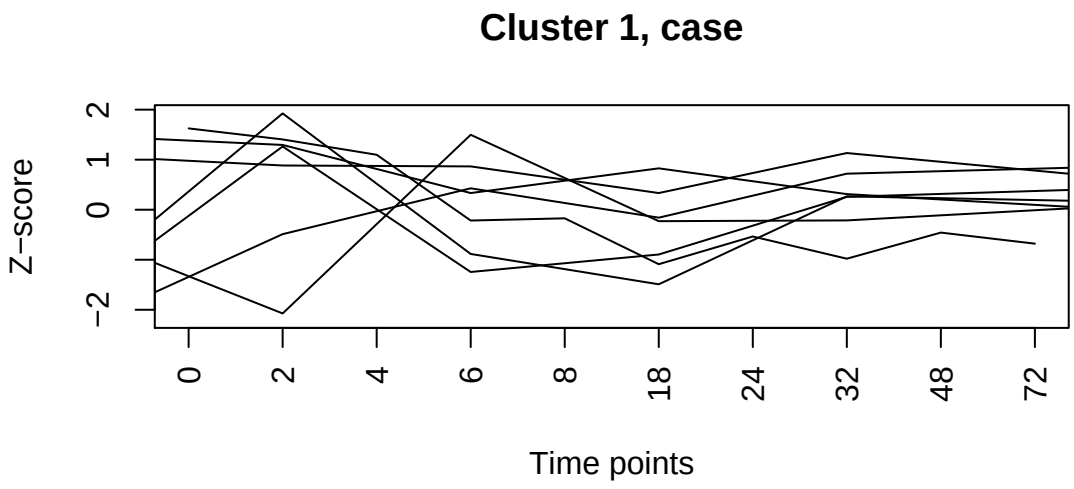
```
# load package
library(ImpulseDE)
```

```
## Loading required package: parallel
```

```
## Loading required package: compiler
```

```
# start analysis
impulse_results <- impulse_DE(t(tcell.10)[1:20,], annot, "Time", "Condition",
   n_iter = 10, n_randoms = 10, n_process = 1, new_device = FALSE)
```

```
## [1] "START: Prepare annotation table for internal usage"
## [1] "-------------------------------------------------------------------"
## [1] "Case condition: activated"
## [1] "DONE"
## [1] "###################################################################"
## [1] "START: Clustering genes for Impulse model fit"
## [1] "-------------------------------------------------------------------"
## [1] "Clustering of case data set"
## [1] "- 13 genes were excluded due to very low variation"
## [1] "--- Number of correlation-based pre-clusters: 1"
## [1] "------ Number of genes in pre-clusters: C1: 7"
## [1] "--- Final number of clusters: 1"
## [1] "------ Number of genes in final clusters: C1: 7"
```




```
## [1] "DONE"
## [1] "Consumed time: 0 min"
## [1] "###################################################################"
## [1] "START: Fitting Impulse model to the clusters"
## [1] "-------------------------------------------------------------------"
## [1] "DONE"
## [1] "Consumed time: 0.02 min"
## [1] "###################################################################"
## [1] "START: Fitting Impulse model to the genes"
## [1] "-------------------------------------------------------------------"
## [1] "DONE"
## [1] "Consumed time: 0.53 min"
## [1] "###################################################################"
## [1] "START: Generate background"
## [1] "-------------------------------------------------------------------"
## [1] "DONE"
## [1] "Consumed time: 0.14 min"
## [1] "###################################################################"
```

```
## [1] "START: DE analysis"
## [1] "-----------------------------------------------------------------------"
## [1] "Found 7 DE genes"
## [1] "DONE"
## [1] "Consumed time: 0 min"
## [1] "#######################################################################"
## [1] "TOTAL consumed time: 0.7 min"
```

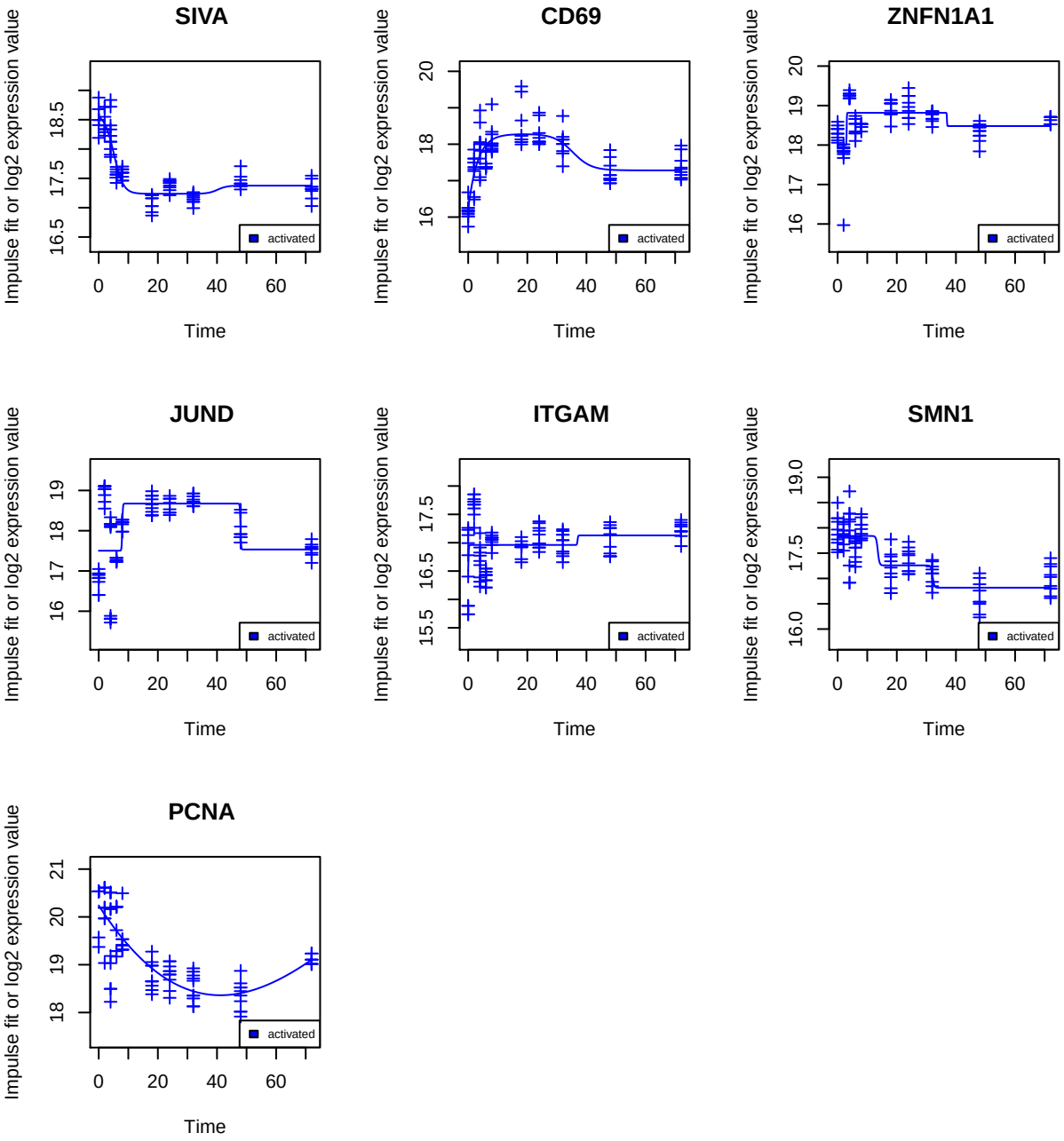
(**Note**: *new_device* is set to *FALSE* in all plot functions here to avoid the generation of emtpy pages within the vignette. Usually, it is recommend to keep this option *TRUE*, which will open a new device for each plot. Otherwise, all earlier plots will be overwritten.)

Plotting genes of interest 1TK

Plotting a custom list of genes can be done by using the function *plot_impulse*. For this the fitting results are needed, which can be taken from the generated result object *impulse_results*. As an example, some genes being called as differentially expressed are plotted.

```
genes = c("SIVA","CD69","ZNFN1A1","JUND","ITGAM","SMN1","PCNA")
plot_impulse(gene_IDs = genes, data_table = t(tcell.10), data_annotation = annot,
    imp_fit_genes = impulse_results$impulse_fit_results,
    file_name_part = "four_NV_genes", new_device = FALSE)
```

```
## [1] "---Plotting genes"
## [1] "SIVA"    "CD69"    "ZNFN1A1" "JUND"    "ITGAM"   "SMN1"    "PCNA"
```



For example, *JUND* and *CD69* show very typical impulse-like expression patterns, which cleary change significantly over time.

Imputing values for missing time points 1TK

To impute values for an uncovered time point for a specific gene, the following command can be used:

```
# impute expression value for time point 60 for gene "JUND"
(imp_results <-
     calc_impulse(impulse_results$impulse_fit_results$impulse_parameters_case[
         "JUND",1:6], 60))
```

```
## [1] 17.53342
```

Differential expression analysis between two time courses

In the case of a two time course experiment, *ImpulseDE* will detect differentially expressed genes between both conditions. In order to generate two time courses out of the T cell data set, the replicates will be splitted and to the second half some random numbers are added:

```
# split dataset into two halfs
case_data <-  t(tcell.10)[,seq(1,ncol(t(tcell.10)),2)]
control_data <-  t(tcell.10)[,seq(2,ncol(t(tcell.10)),2)]
# add some random values to "control_data" to make data different
control_data <- control_data + t(apply(control_data,1,function(x)
    runif(length(x),0,0.5)*sample(c(-1,1),length(x), replace = TRUE)
    + sample(c(seq(-2,2,0.5)),1)))
tcell_2tk <- cbind(case_data, control_data)
```

At last, a proper annotation table has to be generated:

```
annot_2tk <- annot[colnames(tcell_2tk),]
annot_2tk[51:100,"Condition"] = "control"
head(annot_2tk)
```

```
##     Time Condition
## 0-1    0 activated
## 0-3    0 activated
## 0-5    0 activated
## 0-7    0 activated
## 0-9    0 activated
## 2-1    2 activated
```

```
tail(annot_2tk)
```

```
##       Time Condition
## 48-10   48   control
## 72-2    72   control
## 72-4    72   control
## 72-6    72   control
## 72-8    72   control
## 72-10   72   control
```

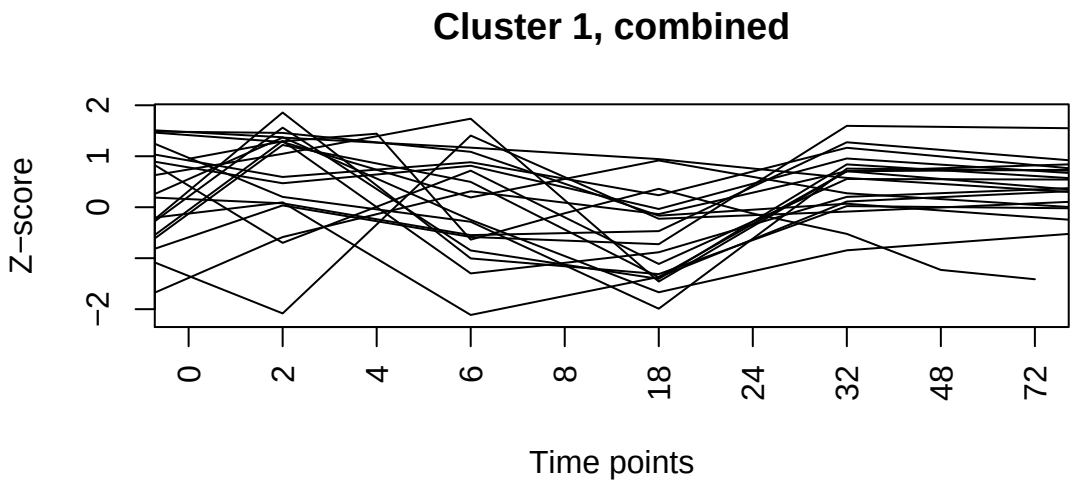
Running the analysis 2TK

In contrast to the single time course scenario, six variables need to be set: *expression_table*, *annotation_table*, *colname_time*, *colname_condition*, *control_timecourse* and *control_name*. Here again, for demonstration purposes the number of itertions, randomizations as well as the number of used processors will be reduced. For real datasets, it not recommended to reduce *n_iter* as well as *n_randoms*. Again, the analysis will be reduced to the first 20 genes:

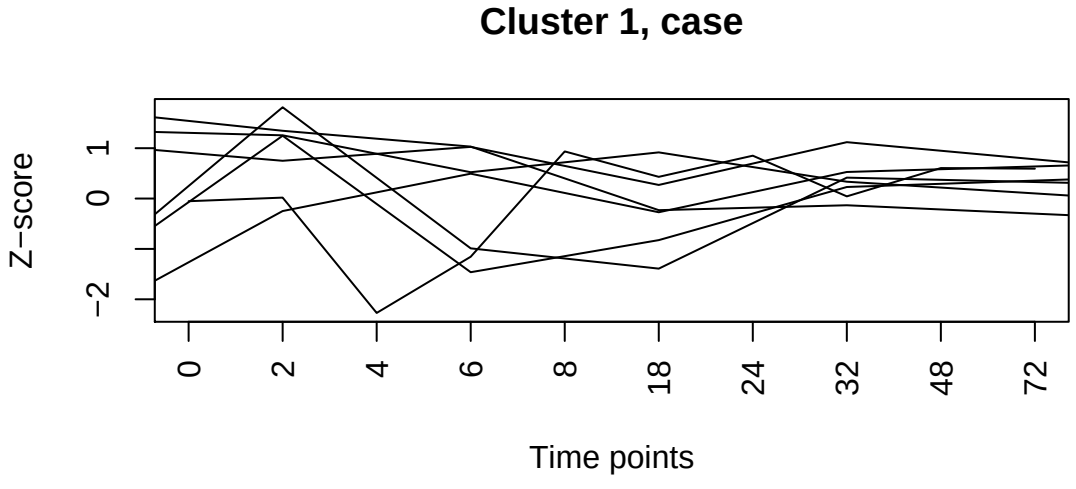
```
# load package
library(ImpulseDE)
# start analysis
impulse_results <- impulse_DE(tcell_2tk[1:20,], annot_2tk, "Time", "Condition",
    TRUE, "control", n_iter = 10, n_randoms = 10, n_process = 1, new_device = FALSE)
```

```
## [1] "START: Prepare annotation table for internal usage"
## [1] "-----------------------------------------------------------------"
## [1] "Case condition: activated"
```

```
## [1] "Control condition: control"
## [1] "DONE"
## [1] "########################################################################"
## [1] "START: Clustering genes for Impulse model fit"
## [1] "------------------------------------------------------------------------"
## [1] "Clustering of combined data set"
## [1] "- 1 genes were excluded due to very low variation"
## [1] "--- Number of correlation-based pre-clusters: 1"
## [1] "------ Number of genes in pre-clusters: C1: 19"
## [1] "--- Final number of clusters: 1"
## [1] "------ Number of genes in final clusters: C1: 19"
```

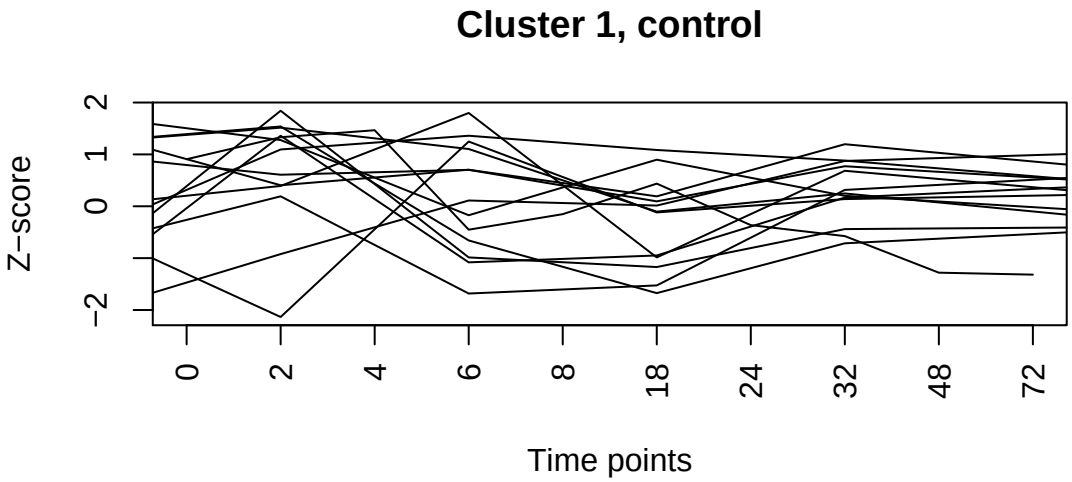



```
## [1] "Clustering of case data set"
## [1] "- 13 genes were excluded due to very low variation"
## [1] "--- Number of correlation-based pre-clusters: 1"
## [1] "------ Number of genes in pre-clusters: C1: 7"
## [1] "--- Final number of clusters: 1"
## [1] "------ Number of genes in final clusters: C1: 7"
```




```
## [1] "Clustering of control data set"
## [1] "- 7 genes were excluded due to very low variation"
## [1] "--- Number of correlation-based pre-clusters: 1"
## [1] "------ Number of genes in pre-clusters: C1: 13"
## [1] "--- Final number of clusters: 1"
```

```
## [1] "------ Number of genes in final clusters: C1: 13"
```




```
## [1] "DONE"
## [1] "Consumed time: 0 min"
## [1] "########################################################################"
## [1] "START: Fitting Impulse model to the clusters"
## [1] "------------------------------------------------------------------------"
## [1] "DONE"
## [1] "Consumed time: 0.05 min"
## [1] "########################################################################"
## [1] "START: Fitting Impulse model to the genes"
## [1] "------------------------------------------------------------------------"
## [1] "DONE"
## [1] "Consumed time: 1.69 min"
## [1] "########################################################################"
## [1] "START: Generate background"
## [1] "------------------------------------------------------------------------"
## [1] "DONE"
## [1] "Consumed time: 0.29 min"
## [1] "########################################################################"
## [1] "START: DE analysis"
## [1] "------------------------------------------------------------------------"
## [1] "Found 19 DE genes"
## [1] "DONE"
## [1] "Consumed time: 0 min"
## [1] "########################################################################"
## [1] "TOTAL consumed time: 2.04 min"
```

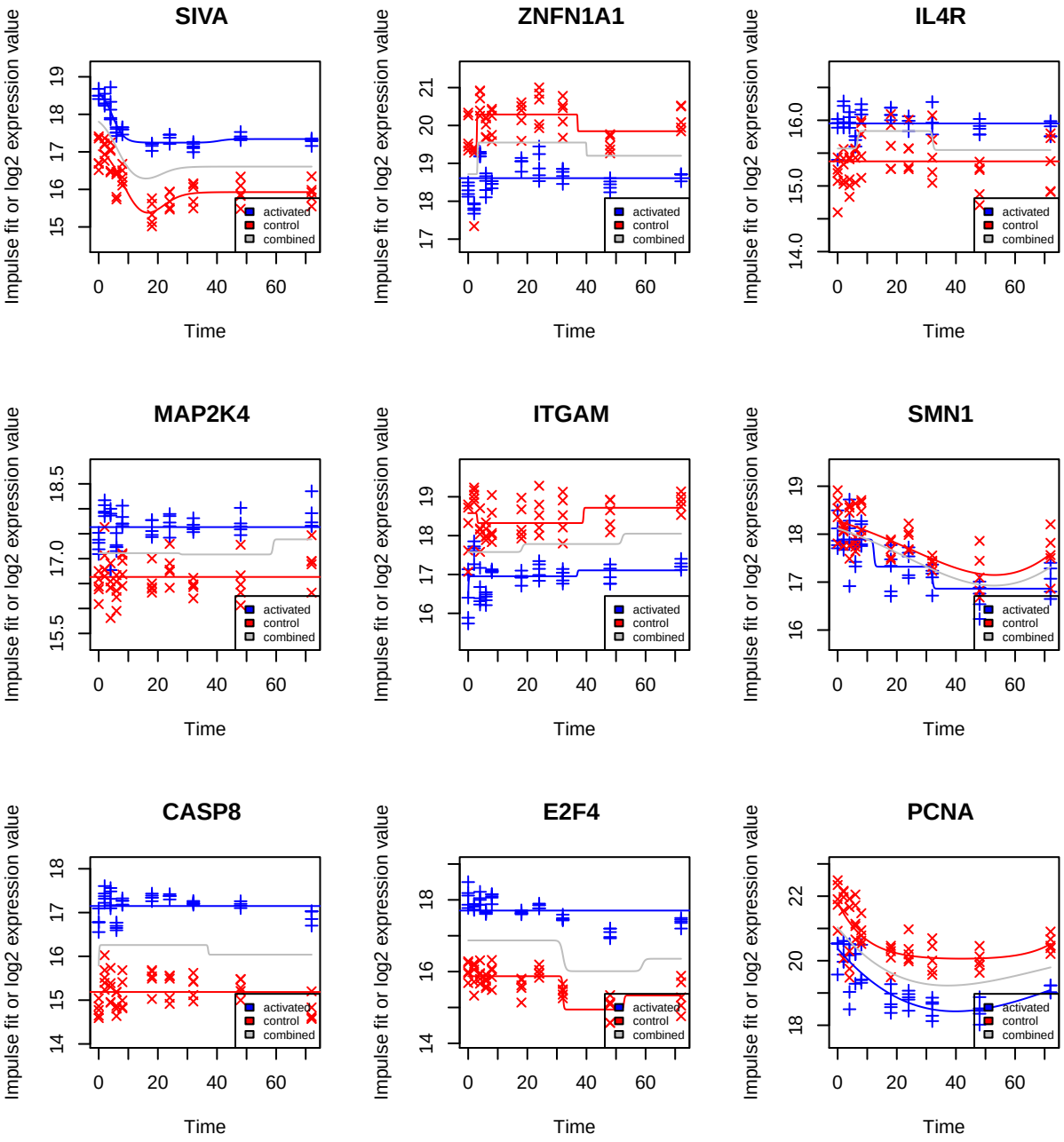
Plotting genes of interest 2TK

As an example, some genes being called as differentially expressed are plotted as well:

```
genes = c("SIVA","ZNFN1A1","IL4R","MAP2K4","ITGAM","SMN1","CASP8","E2F4","PCNA")
plot_impulse(gene_IDs = genes, data_table = tcell_2tk, data_annotation =
    annot_2tk, imp_fit_genes = impulse_results$impulse_fit_results,
    control_timecourse = TRUE,
    control_name = "control", file_name_part = "four_NV_genes_2tk", new_device = FALSE)
```

```
## [1] "---Plotting genes"
```

```
## [1] "SIVA"    "ZNFN1A1" "IL4R"    "MAP2K4"  "ITGAM"   "SMN1"    "CASP8"
## [8] "E2F4"    "PCNA"
```



Imputing values for missing time points 2TK

To impute values for an uncovered time point for a specific gene, *calc_impulse* has to be applied to both datasets separately:

```
# impute expression value for time point 60 for gene "JUND"
# case data
(imp_results <-
     calc_impulse(impulse_results$impulse_fit_results$impulse_parameters_case[
         "JUND",1:6], 60))
```

```
## [1] 17.52859
```

```
# control data
(imp_results <-
    calc_impulse(impulse_results$impulse_fit_results$impulse_parameters_control[
    "JUND",1:6], 60))
```

```
## [1] 16.99018
```

References

Benjamini, Yoav, and Yosef Hochberg. 1995. "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing." *Journal of the Royal Statistical Society Series B (Methodological)* 57 (1):289–300. https://doi.org/10.2307/2346101.

Chechik, Gal, and Daphne Koller. 2009. "Timing of Gene Expression Responses to Environmental Changes." *Journal of Computational Biology* 16 (2):279–90. https://doi.org/10.1089/cmb.2008.13TT.

Rangel, Claudia, John Angus, Zoubin Ghahramani, Maria Lioumi, Elizabeth Sotheran, Alessia Gaiba, David L. Wild, and Francesco Falciani. 2004. "Modeling T-Cell Activation Using Gene Expression Profiling and State-Space Models." *Bioinformatics* 20 (9):1361–72. https://doi.org/10.1093/bioinformatics/bth093.

Storey, John D., Wenzhong Xiao, Jeffrey T. Leek, Ronald G. Tompkins, and Ronald W. Davis. 2005. "Significance Analysis of Time Course Microarray Experiments." *Proceedings of the National Academy of Sciences* 102 (36):12837–42. https://doi.org/10.1073/pnas.0504609102.