Fractional response analysis - R package

User Manual

K. Nienałtowski*, R.E. Rigby, J.Walczak, K.Zakrzewska, J.Rehwinkel, M. Komorowski

17 czerwca 2020

Abstract

R- package FRA is designed to perform fractional response analysis of single-cell responses as presented in the manuscript Nienałtowski et al. "Fractional response analysis reveals logarithmic cytokine responses in cellular populations".

Contents

*author and maintainer, please contact via karol.nienaltowski a t gmail.com

Preliminaries

Requirements - Hardware

- A 32 or 64 bit processor (recommended: 64bit)
- 1GHz processor (recommended: multicore for a comprehensive analysis)
- 2GB MB RAM (recommended: 4GB+, depends on the size of experimental data)

Requirements - Software

The main software requirement is the installation of the R environment (version: >= 3.2), which can be downloaded from R project website and is distributed for all common operating systems. We tested the package in R environment installed on Windows 7, 10; Mac OS X 10.11 - 10.13 and Ubuntu 18.04 with no significant differences in the performance. The use of a dedicated Integrated development environment (IDE), e.g. RStudio is recommended.

Apart from a base installation of R, FRA requires the following R packages:

1. for installation

- devtools

2. for estimation

- nnet
- doParallel (if parallel computation are needed)

3. for visualisation

- ggplot2
- ggthemes
- grDevices
- viridis

4. for data handling

- data.table
- reshape2
- dplyr
- foreach

Each of the above packages can be installed by executing

```
install.packages("name_of_a_package")
```

in the R console.

Importantly, during installation availability of the above packages will be verified and missing packages will be automatically installed.

Installation

The package can be directly installed from GitHub. For installation, open RStudio (or base R) and run following commands in the R console

```
install.packages("devtools") # run if 'devtools' is not installed
library(devtools)
install_github("sysbiosig/FRA")
```

All packages that are required will be installed or updated automatically.

Citing and support

The package implements methods published, please cite:

Nienałtowski K, Rigby R.E., Walczak J., Zakrzewska K.E., Rehwinkel J, and Komorowski M (2020) Fractional response analysis reveals logarithmic cytokine responses in cellular populations.

All problems, issues and bugs can be reported here:

https://github.com/sysbiosig/FRA/issues

or directly via e-mail: karol.nienaltowski@gmail.com.

Package structure

The FRA package provides their functionalities with three main functions:

1. `FRA()`- fractional response analysis performed for heterogeneous, multivariate, and dynamic measurements. Function computes: (i) the fractional response curve (FRC) that quantifies fractions of cells that exhibit different responses to a change in dose, or any other experimental conditionand and (ii) the cell-to-cell heterogeneity, i.e., fraction of cells exposed to one dose that exhibits responses in the range characteristic for other doses.
2. `plotHeterogeneityPieCharts()` - visualises the cell-to-cell heterogeneity structure using table of pie charts. Each pie chart describes the fraction of cells exposed to one dose (rows) that expibits responses typical for either of the doses (columns).
3. `plotFRC()` - visualises FRC and the cell-to-cell heterogenity. FRC is represented as a line, whereas heterogeneity is represented as colour band.

Morevoer, package contains examplary datasets, that were used in the publication:

1. `data.fra.cytof`
2. `data.fra.ps1`
3. `data.fra.ps3`
4. `data.fra.nfkb`

Input data

The function `FRA()` takes `data` in the form of the object `data.frame` with a specifc structure of rows and columns. Responses y_j^i are assumed to be measured for a finite set of stimuli levels $x_1, x_2, \ldots, x_m$. The responses y_j^i can be multidimensional. Usually, experimental dataset is represented as a table with rows and columns organized as shown in Figure 1.

input	output 1	output 2	output 3	...
n_1 $\left\{\begin{array}{l} x_1 \\ \vdots \\ x_1 \end{array}\right.$	$\begin{array}{c} y^1_{1,1} \\ \vdots \\ y^1_{n_1,1} \end{array}$	$\begin{array}{c} y^1_{1,2} \\ \vdots \\ y^1_{n_1,2} \end{array}$	$\begin{array}{c} y^1_{1,m} \\ \vdots \\ y^1_{n_1,m} \end{array}$	
n_2 $\left\{\begin{array}{l} x_2 \\ \vdots \\ x_2 \end{array}\right.$	$\begin{array}{c} y^2_{1,1} \\ \vdots \\ y^2_{n_2,1} \end{array}$	$\begin{array}{c} y^2_{1,2} \\ \vdots \\ y^2_{n_2,2} \end{array}$	$\begin{array}{c} y^2_{1,m} \\ \vdots \\ y^2_{n_2,m} \end{array}$	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	...
n_m $\left\{\begin{array}{l} x_m \\ \vdots \\ x_m \end{array}\right.$	$\begin{array}{c} y^m_{1,1} \\ \vdots \\ y^m_{n_m,1} \end{array}$	$\begin{array}{c} y^m_{1,2} \\ \vdots \\ y^m_{n_m,2} \end{array}$	$\begin{array}{c} y^m_{1,m} \\ \vdots \\ y^m_{n_m,m} \end{array}$	

Figure 1: Structure of an experimental dataset required for running FRA

Data example

An example of the input `data.frame`, which contains the multivariate dose-responses to IFN-a2a in **monocytes CD14+** presented in the **MP** is available within the package under the variable `FRA::data.scrc.cytof`. It has the following format

Stim	pSTAT1	pSTAT3	pSTAT4	pSTAT5	pSTAT6
25	9.423005	0.0000000	0.000000	0.2226261	1.0420985
25	14.077361	0.0000000	0.000000	4.4516195	0.0000000
25	6.634479	2.3057582	0.000000	0.0000000	0.8014894
25	5.739593	0.0000000	3.807480	0.3100144	0.6536241
25	24.686699	0.0000000	0.000000	0.1629745	1.2835708
25	21.104772	0.3036974	1.324072	3.2628903	0.6757941

where each row represents measurements of a single-cell, the column named `Stim` specifies the dose level of IFN-a2a, while `pSTAT1`, `pSTAT3`, `pSTAT4`,`pSTAT5`,`pSTAT6` are the normalized levels of phosporylated STATs in an individual cell. The above table can be shown in R by calling

```
head(FRA::data.itrc.cytof)
```

Basic usage

Then, main function is called as:

```
model <-  FRA::FRA(
  data = data,
  signal = "input",
  response = c("output_1", "output_2", "output_3", ...),
  bootstrap.number = bootstrap.number,
  ...
)
```

Variables `signal` and `response` describes respectively dose level and single-cell responses. These columns should be of type `numeric`; order and number of outputs should be the same for all cells. Number of observations in `data` shoulb large, possibly >100, per input value is required.

The variable `bootstrap.number` represents number of bootstrap samples required for estimation of cell-to-cell heterogeneity. It is crucial to choose this value carefully, as it induce estimator accuracy.

The result of the function is an object of class `FRAModel`, that contains results of the estimator. To see the results call :

```
print(model)
```

To get FRA (cumulative frequency) and cell-to-cell heterogeneity please call `model$frc` and `model$heterogeneity`, respectively.

Results can be visualised using one of our plots, as it was presented in the publication.

FRA can be plotted using function:

```
plotFRC(model)
```

The cell-to-cell heterogeneity can be plotted using function:

```
plotHeterogeneityPieCharts(model)
```

Example

Below, we present an application of `FRA` package to the case of the multivariate dose-responses to IFN-a2a in **monocytes CD14+** described in the article. Fractional response analysis are computed by calling function:

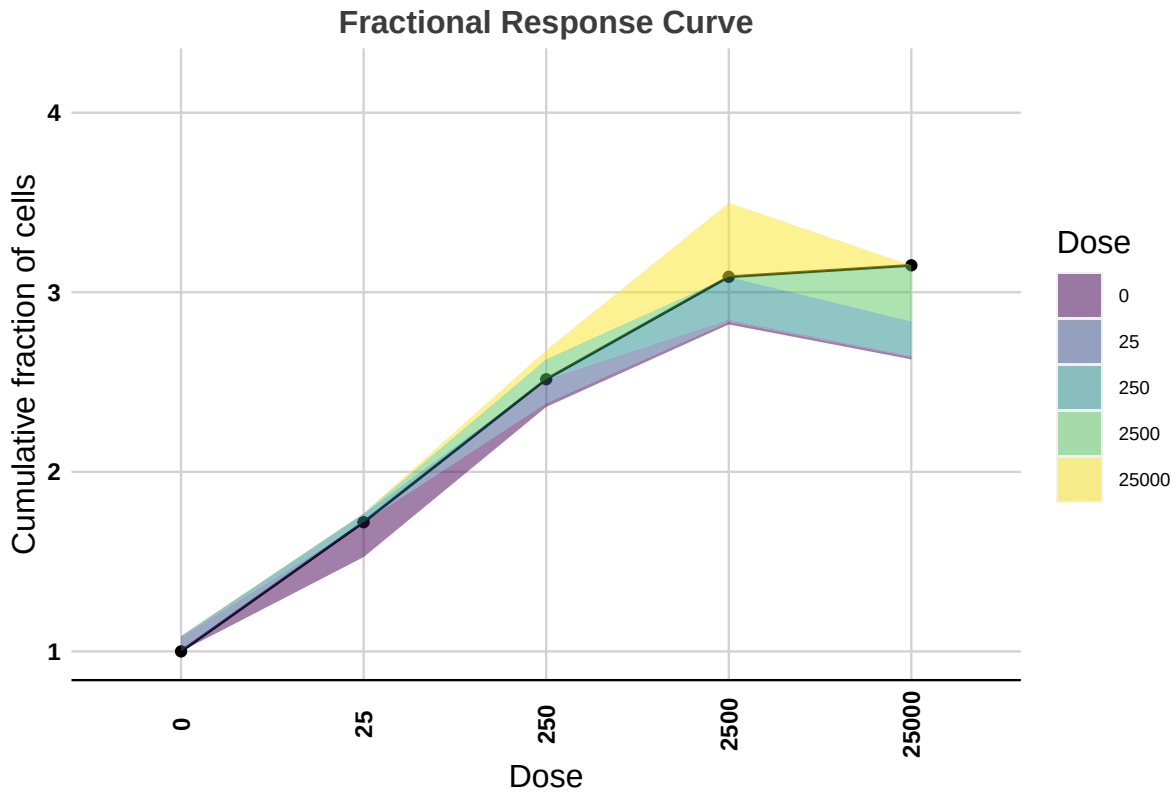
The result is called by:

```
print(model)
```

```
## FRAModel
## formula : Stim ~ pSTAT1+pSTAT3+pSTAT4+pSTAT5+pSTAT6
## FRA :
##     0    25   250  2500 25000
##  1.00  1.72  2.52  3.09  3.15
## confusion matrix :
##          0   25  250 2500 25000
## 0     0.91 0.08 0.00 0.00  0.00
## 25    0.19 0.76 0.05 0.00  0.00
## 250   0.02 0.13 0.68 0.11  0.05
## 2500  0.02 0.01 0.24 0.32  0.41
## 25000 0.01 0.01 0.19 0.31  0.48
```

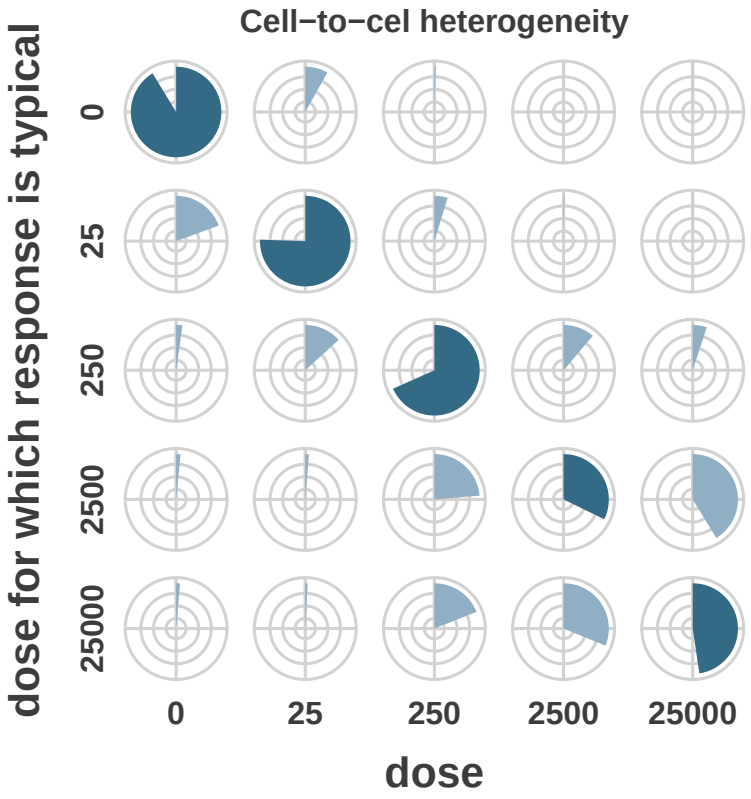
To plot fractional response curve is plotted by calling

```
FRA::plotFRC(model = model)
```



To obtain the cell-to-cell heterogeneity as a pie charts call:

```
FRA::plotHeterogeneityPieCharts(model = model)
```



Details of FRA packgae functions

Fractional response analysis

In order to perform fractional response analysis of single-cell data call

```
model <-FRA(
  data,
  signal = "signal",
  response = "response",
  sample = "sample",
  bootstrap.number = 0,
  bootstrap.sample_size = 1000,
  parallel_cores = 1,
  lr_maxit = 1000,
  MaxNWts = 5000,
  ...
)
```

- `data` - a data.frame or data.table object in a wide format that describe response (might be multi-dimmensional) of the samples to the signal (now only one dimmensional); data.frame data consists columns of names defined by sample, signal (optional), and response; each row represents a response of one sample to the input signal; column signal define the input signal; columns response define the multidimmensional (optional) response to the input signal; column sample specify identifaction of sample; if sample is not defined then sample is identified by row number;
- `signal` - character, specify name of the column that represents the input signal;

- `response` vector of characters, that specify names of the columns that represents the output response;
- `sample` - character (optional), specify name of the column that consists identifiaction of sample;
- `parallel_cores` - specify number of cores used for computations, `default = 1`
- `bootstrap.number` (`default = 1`) - numeric, `bootstrap.number >= 1`, specify nymber of bootstrap samples used for estimation SCRC and cell-to-cell heterogeneity. It is crucial to choose this value carefully, as it induce estimator accuracy. The proper value depends on data dimmensions and density distribution. The practice indicates that the higher number of bootstrap samples are required to obtain satisfying level of the accuracy of the cell-to-cell heterogeneity estimator. The `bootstrap.number = 1` denotes that one bootstrap sampling is performed to guarantee equipotence between number of cells for each dose, that is assumed in method;
- `bootstrap.sample_size` - numeric, size of the bootstrap sample;
- `lr_maxit` (`default = 1000`) - a maximum number of iterations of fitting step of logistic regression algorithm in `nnet` function. If a warning regarding lack of convergence of logistic model occurs, should be set to a larger value (possible if data is more complex or of a very high dimension);
- `MaxNWts` (`default = 5000`) - a maximum number of parameters in logistic regression model. A limit is set to prevent accidental over-loading the memory. It should be set to a larger value in case of exceptionally high dimension of the output data or very high number of input values. In principle, logistic model requires fitting $(m-1)\cdot(d+1)$ parameters, where m is the number of unique input values and d is the dimension of the output.

The function returns the `FRAModel` object that contains among others * `frc` - a `data.frame` that describe fractional response curve; contains two columns `dose` and `frc` * `heterogeneity` - a `data.frame` that describes cell-to-cell heterogeneity, i.e., fraction of cells exposed to one dose (rows) that exhibits responses in the range characteristic for other doses (columns).

plotFRC

In order to visualise fractional response curve call

```
plotFRC(
    model,
    title_ =
      "Fractional Response Curve",
    xlab_ = "Dose",
    ylab_ = "Cumulative fraction of cells",
    fill.guide_ = "legend",
    ylimits_ = TRUE,
    alpha_ = 0.5,
    theme.signal = NULL,
    plot.heterogeneity = TRUE,
    ...
```

- `model` - FRAModel object return by FRA function
- `title_` - character, specify title of plot, default "Fractional Response Curve"
- `xlab_` - character, label of x axes, default "Dose"
- `ylab_` - character, label of y axes and legend title, default "Cumulative fraction of cells"
- `fill.guide_` - argument specify if legend should be displayed; legend is displayed for `fill.guide_ = "legend"`, legend is not displayed for `fill.guide_ = NULL`, `default = "legend"`
- `ylimits_` - logical (TRUE or FALSE) or numeric vector of minimum and maximum of y axes,
- `theme.signal` - optional, object returned by GetRescaledSignalTheme
- `plot.heterogeneity` - logical, define if FRC visualise the heterogeneity structure, `default = TRUE`

plotHeterogeneityPieCharts

In order to visualise th cell-to-cell heterogeneity structure, call

```
plotHeterogeneityPieCharts(
  model,
  max.signal = NULL,
  title_ = "Cell-to-cel heterogeneity",
  ylab_ = "dose",
  xlab_ = "dose for which response is typical",
  ...
)
```

- `model` - FRAModel object return by FRA function
- `max.signal` - maximal signal for which the cell-to-cell heterogeneity is plotted, `default = max(signal)` *`title_` - character, specify title of plot, `default = "Cell-to-cel heterogeneity"`
- `ylab_` - character, label of y axes, `default = "dose"`
- `xlab_` - character, label of x axes, `default = "dose for which response is typical"`