

# Package ‘PGPC’

April 12, 2018

**Type** Package

**Title** Experimental data and analysis of the chemical-genetic interaction screen in isogenic HCT116 cell lines

**Version** 1.6.0

**Author** Felix Klein

**Maintainer** Felix Klein <daggioth@gmx.de>

**Description** This package contains the experimental data and a vignette guiding through the analysis of a chemical-genetic interaction screen in isogenic HCT116 cell lines. The code can be executed to generate all results and figures for the manuscript "A chemical-genetic interaction map of small molecules using high-throughput imaging in cancer cells" accepted for publication at Molecular Systems Biology. Data availability: Complementary views on this dataset are available through different repositories. The image data files are available from the BioStudies database at the European Bioinformatics Institute (EMBL-EBI) under the accession S-BSMS-PGPC1 (<http://wwwdev.ebi.ac.uk/biostudies/studies/S-BSMS-PGPC1>) An interactive front-end for exploration of the images is provided by the IDR database <http://dx.doi.org/10.17867/10000101>. The authors are hosting an interactive webpage to browse images and interaction profiles at <http://dedomena.embl.de/PGPC>.

**License** Artistic-2.0

**LazyLoad** true

**Depends** R (>= 3.0), EBImage, imageHTS, SearchTrees, limma, RColorBrewer, gplots, splots, ggplot2, geneplotter, ChemmineR, reshape2, plyr

**VignetteBuilder** knitr

**Suggests** BiocStyle, knitr

**biocViews** CancerData, ColonCancerData, ExperimentData

**NeedsCompilation** no

## R topics documented:

datamatrixTransformed . . . . .	2
ftsr . . . . .	3
getFeaturesAllSpots . . . . .	3

getInteractions . . . . .	4
interactions . . . . .	6
mergeProfiles . . . . .	7
segmentAllSpots . . . . .	8
segmentXman . . . . .	9
selected . . . . .	10
summarizeWellsExtended . . . . .	11

<b>Index</b>	<b>13</b>
--------------	-----------

---

datamatrixTransformed *Intermediate data of the analysis pipeline*

---

## Description

Glog transformed feature data represented as a list containing array D and list anno.

## Usage

```
data(datamatrixTransformed)
```

## Format

```
List of 2
 $ D   : num [1:1372, 1:12, 1:2, 1:385]
 $ anno:List of 4
 ..$ drug:'data.frame': 1372 obs. of  15 variables
 ..$ line:'data.frame': 12 obs. of  4 variables
 ..$ repl: int [1:2]
 ..$ ftr : chr [1:385]
```

## Value

The four dimensional array D contains the glog transformed feature data which were rearranged from the original ftrs data.frame. The dimensions represent:

1. drug
2. cell line
3. replicate
4. feature

The annotation anno is represented as a list containing a data.frame drug with the drug annotation, a data.frame line with the cell line annotation, a vector repl with the information about the replicates and a vector ftr with the feature names.

## Examples

```
data(datamatrixTransformed)
str(datamatrixTransformed)
```

---

`ftrs`*Well profiles extracted from images using imageHTS*

---

**Description**

This file contains the data.frame output of the imageHTS pipeline.

**Usage**

```
data(ftrs)
```

**Format**

A data frame with 36864 observations on 396 variables. The first is 'uname', a character vector of the well unames. The remaining 395 variables are numeric vectors of feature summaries for each well.

**Value**

Feature summaries calculated over all features extracted from the cells in each well. The wells are defined by their uname.

**See Also**

[extractFeatures](#), [computeFeatures](#), [getFeaturesAllSpots](#), [summarizeWellsExtended](#)

**Examples**

```
data(ftrs)
names(ftrs)
```

---

`getFeaturesAllSpots`*Function used to segment all spots for a well*

---

**Description**

`getFeaturesAllSpots` is called by the function `extractFeatures` to segment the images of the screen

**Usage**

```
getFeaturesAllSpots(cal, seg, p)
```

**Arguments**

<code>cal</code>	Calibrated RGB image matrix.
<code>seg</code>	List of the nuclear and cell segmentation masks.
<code>p</code>	List of parameters which are read from the file specified in <code>featurePar</code> of the <code>extractFeatures</code> function.

**Value**

Returns a data.frame with the extracted features.

**Author(s)**

Felix A. Klein, <felix.klein@embl.de>

**See Also**

[imageHTS](#), [extractFeatures](#)

**Examples**

```
## see section 2.1 Image processing on cluster for a working example
localPath <- tempdir()
serverURL <- system.file("extdata", package = "PGPC")

imageConfFile = file.path("conf", "imageconf.txt")

## circumvent memory problem on 32bit windows by segmenting only spot 1.
if(.Platform$OS.type == "windows" & R.Version()$arch == "i386")
  imageConfFile = file.path("conf", "imageconf_windows32.txt")

x = parseImageConf(imageConfFile,
                   localPath=localPath,
                   serverURL=serverURL)

well <- "045-01-C23"

## get segmentation parameter
p <- readHTS(x, type = "file",
            filename = file.path("conf", "segmentationpar.txt"),
            format = "dcf")

segmentation <- segmentAllSpots(x, well, p, access="cache")

## get feature parameter
pf <- readHTS(x, type = "file",
            filename = file.path("conf", "featurepar.txt"),
            format = "dcf")

ftrs <- getFeaturesAllSpots(cal=segmentation$cal,
                          seg=list(nseg=segmentation$nseg,
                                   cseg=segmentation$cseg),
                          pf)
```

## Description

To detect chemical genetic interactions, the data of each feature is modeled using a multiplicative model and robust L1 regression to estimate the effects of the cell line and drug treatment using the `medpolish` function. In this iterative approach row and column median values are subtracted alternately until the proportional change of the absolute residuals falls below a defined threshold. The final row and column values describe the drug and cell line effect respectively. The residuals represent the interaction terms. This process is done for each replicate and each feature individually. To detect significant interactions the values of replicates are used to perform a moderated t-test against the null hypothesis  $t = 0$ , using the implementation in the Bioconductor package `limma`. p-values are adjusted for multiple testing by controlling for the false discovery rate using the method of Benjamini & Hochberg.

## Usage

```
getInteractions(d, ftrs = NULL, samplesOnly = FALSE, scaleByLine = FALSE,
  ...)
```

## Arguments

<code>d</code>	list containing an array with four dimensions of features <code>D</code> , a list with annotation. The annotation list needs to contain a character vector <code>ftr</code> with the feature names and a character vector <code>drug\$Content</code> defining whether the data comes from "sample" or "other" wells.
<code>ftrs</code>	Parameter to select certain features for the calculation of chemical genetic interactions.
<code>samplesOnly</code>	If set to TRUE only values of "sample" wells will be used for the calculation of chemical genetic interactions.
<code>scaleByLine</code>	If set to TRUE the interaction terms for each cell line will be scaled by the median absolute deviation of interaction terms for the individual cell line and replicate.
<code>...</code>	Additional parameters passed to <code>medpolish</code>

## Value

A list with the annotation `anno`, raw data of the selected features `D`, the chemical genetic interaction results `res`, the estimated drug and cell line effect `effect` and calculated p-values and multiple testing adjusted p-values.

## Author(s)

Felix A. Klein, <felix.klein@embl.de>

## See Also

[medpolish](#), [interactions](#), [lmFit](#), [eBayes](#)

## Examples

```
data(interactions)
x <- getInteractions(interactions)
```

---

interactions

*Interaction data generated by the analysis pipeline*

---

## Description

This file contains the result of the analysis pipeline calling `getInteractions`.

## Usage

```
data(interactions)
```

## Format

```
List of 5
 $ anno :List of 4
   ..$ drug:'data.frame': 1372 obs. of  15 variables
   ..$ line:'data.frame': 12 obs. of  4 variables
   ..$ repl: int [1:2]
   ..$ ftr : chr [1:20]
 $ D     : num [1:1372, 1:12, 1:2, 1:20]
 $ res   : num [1:1372, 1:12, 1:2, 1:20]
 $ effect:List of 2
   ..$ drug: num [1:1372, 1:2, 1:20]
   ..$ line: num [1:12, 1:2, 1:20]
 $ pVal  : num [1:1372, 1:12, 1:20, 1:3]
```

## Value

`interactions` is a list containing the list `anno`, array `D`, array `res`, list `effect` and array `pVal`.

The annotation `anno` is represented as a list containing a `data.frame` `drug` with the drug annotation, a `data.frame` `line` with the cell line annotation, a vector `repl` with the information about the replicates and a vector `ftr` with the feature names.

The four dimensional array `D` contains the glog transformed feature data of the features selected for the final analysis. The dimensions represent:

1. drug
2. cell line
3. replicate
4. feature

The four dimensional array `res` contains the interaction terms. It has the same dimensions as `D`. The dimensions represent:

1. drug
2. cell line
3. replicate
4. feature

The list effect contains the drug and cell line effect as three-dimensional array: drug and line have the following dimensions:

1. drug or cell line respectively
2. replicate
3. feature

pVal is an array containing the p-values, adjusted p-values and correlation between replicates of interactions. The dimensions represent:

1. drug
2. cell line
3. p-value, adjusted p-value, correlation
4. feature

### See Also

[getInteractions](#)

### Examples

```
data(interactions)
str(interactions)
```

---

mergeProfiles	<i>Function to merge profiles of extracted features from parallel processing</i>
---------------	--

---

### Description

Merges all feature profiles of each well and saves the result in the specified file.

### Usage

```
mergeProfiles(x, profilename = "profiles", output = "profiles.tab",
  folder = "data", access = "cache")
```

### Arguments

x	A imageHTS object.
profilename	pattern of profile file names.
output	File name to save the merged profiles.
folder	folder name in which the profile files are stored and in which the the result is saved.
access	Access parameter passed to fileHTS

### Value

None, writes the merged profiles into the specified file on disk.

**Author(s)**

Felix A. Klein, <felix.klein@embl.de>

**See Also**

[imageHTS](#), [extractFeatures](#), [fileHTS](#)

**Examples**

```
## see section 2.1 Image processing on cluster for usage
```

---

segmentAllSpots	<i>Function used to segment all spots for a well</i>
-----------------	--

---

**Description**

segmentAllSpots is called by the imageHTS function segmentWells to segment the images of the screen

**Usage**

```
segmentAllSpots(x, uname, p, access)
```

**Arguments**

x	A imageHTS object.
uname	A character vector, containing the well names to segment. See getUnames for details.
p	List of parameters which are read from the file specified in segmentationPar of the segmentWells function.
access	A character string indicating how to access the data. Valid values are 'local', 'server' and 'cache', the default. See fileHTS for details.

**Value**

Returns a list with the following items cal: calibrated RGB image of the different channels nseg: nuclear segmentation mask cseg: cell segmentation mask

**Author(s)**

Felix A. Klein, <felix.klein@embl.de>

**See Also**

[imageHTS](#), [segmentWells](#), [fileHTS](#)



**Examples**

```

## see section 2.1 Image processing on cluster for a working example
localPath = tempdir()
serverURL = system.file("extdata", package = "PGPC")

imageConfFile = file.path("conf", "imageconf.txt")

## circumvent memory problem on 32bit windows by segmenting only spot 1.
if(.Platform$OS.type == "windows" & R.Version()$arch == "i386")
  imageConfFile = file.path("conf", "imageconf_windows32.txt")

x = parseImageConf(imageConfFile,
  localPath=localPath,
  serverURL=serverURL)

well = "045-01-C23"

## get segmentation parameter
p = readHTS(x, type = "file",
  filename = file.path("conf", "segmentationpar.txt"),
  format = "dcf")

segmentation = segmentAllSpots(x, well, p, access="cache")

```

segmentXman

*Function used to segment a single image***Description**

segmentXman is called by the function segmentAllSpots to segment the single image of one spot.

**Usage**

```
segmentXman(x, uname, p, access, spot = NULL)
```

**Arguments**

x	A imageHTS object.
uname	A character vector, containing the well names to segment. See getUnames for details.
p	List of parameters which are read from the file specified in segmentationPar of the segmentWells function.
access	A character string indicating how to access the data. Valid values are 'local', 'server' and 'cache', the default. See fileHTS for details.
spot	An single integer, indicating the spot number to segment. If it is specified, the default 1 will automatically be used.

**Value**

Returns a list with the following items cal: calibrated RGB image of the different channels nseg: nuclear segmentation mask cseg: cell segmentation mask

**Author(s)**

Felix A. Klein, <felix.klein@embl.de>

**See Also**

[segmentAllSpots](#), [getUnames](#)

**Examples**

```
## see also section 2.1 "Image processing on cluster"
localPath <- tempdir()
serverURL <- system.file("extdata", package = "PGPC")

x <- parseImageConf(file.path("conf", "imageconf.txt"),
                    localPath=localPath,
                    serverURL=serverURL)

well <- "045-01-C23"

## get segmentation parameter
p <- readHTS(x, type = "file",
            filename = file.path("conf", "segmentationpar.txt"),
            format = "dcf")

segmentation <- segmentXman(x, well, p, access="cache", spot=1)
```

---

selected

*Result of the feature selection function*

---

**Description**

This file contains the result of the feature selection function for 40 iterations.

**Usage**

```
data(selected)
```

**Format**

```
List of 4
 $ selected      : chr [1:40]
 $ correlation   : num [1:40]
 $ ratioPositive : num [1:40]
 $ correlationAll:List of 40 Named num vectors
```

**Value**

A list containing the vector `selected` of selected features, the vector `correlation` of their correlations at the time they were selected, the vector `ratioPositive` with the fraction of positive correlations for each iteration and a list `correlationAll` which contains the correlations of all features at each iteration step.

**Examples**

```
data(selected)
str(selected)
```

---

```
summarizeWellsExtended
```

*Function to summarize the extracted features per cell for each well*

---

**Description**

The function extends the imageHTS function `summarizeWells`. It calculates summary statistics over all cells for each well. In particular the trimmed mean (`trim = 0.1`) and `sd` is calculated for each extracted feature. Additionally the `1` calculated for features that are not the standard deviation, median absolute deviation or Halraick statistics calculated over each cell.

**Usage**

```
summarizeWellsExtended(x, uname, featurePar,
  profileFilename = file.path("data", "profiles.tab"), access = "cache")
```

**Arguments**

<code>x</code>	A imageHTS object.
<code>uname</code>	A character vector, containing the well names that will be summarized.
<code>featurePar</code>	File containing the feature parameters used for summarizing the wells.
<code>profileFilename</code>	File name to save the summarized features.
<code>access</code>	Access parameter passed to <code>fileHTS</code>

**Value**

None, writes the summarized well profiles into the specified files on disk.

**Author(s)**

Felix A. Klein, <felix.klein@embl.de>

**See Also**

[imageHTS](#), [summarizeWells](#)

**Examples**

```
## see section 2.1 Image processing on cluster for a working example
localPath = tempdir()
serverURL = system.file("extdata", package = "PGPC")

imageConfFile = file.path("conf", "imageconf.txt")

## circumvent memory problem on 32bit windows by segmenting only spot 1.
if(.Platform$OS.type == "windows" & R.Version()$arch == "i386")
```



# Index

## \*Topic **datasets**

- datamatrixTransformed, [2](#)
- ftrs, [3](#)
- interactions, [6](#)
- selected, [10](#)

computeFeatures, [3](#)

datamatrixTransformed, [2](#)

eBayes, [5](#)

extractFeatures, [3](#), [4](#), [8](#)

fileHTS, [8](#)

ftrs, [3](#)

getFeaturesAllSpots, [3](#), [3](#)

getInteractions, [4](#), [7](#)

getUnames, [10](#)

imageHTS, [4](#), [8](#), [11](#)

interactions, [5](#), [6](#)

lmFit, [5](#)

medpolish, [5](#)

mergeProfiles, [7](#)

segmentAllSpots, [8](#), [10](#)

segmentWells, [8](#)

segmentXman, [9](#)

selected, [10](#)

summarizeWells, [11](#)

summarizeWellsExtended, [3](#), [11](#)