# Package 'OperaMate'

April 23, 2016

**Version** 1.2.3

**Date** 2015-12-28

**Title** An R package of Data Importing, Processing and Analysis for Opera High Content Screening System

**Author** Chenglin Liu

**Maintainer** Chenglin Liu <cliu@sjtu.edu.cn>

**Depends** R (>= 3.2.0),stats,methods,grDevices

**Imports** pheatmap,grid,ggplot2,fBasics,gProfileR,gridExtra,reshape2,stabledist

**Suggests** BiocStyle

**Description** OperaMate is a flexible R package dealing with the data generated by PerkinElmer's Opera High Content Screening System. The functions include the data importing, normalization and quality control, hit detection and function analysis.

**License** GPL (>= 3)

**biocViews** Preprocessing, CellBasedAssays, Normalization, QualityControl

**NeedsCompilation** no

**RoxygenNote** 5.0.1

## R topics documented:

## `cellData-class` *The cellData class*

### Description

The main class used in `OperaMate` to hold all levels of experiment data of a specific type.

### Usage

```
cellData(name, positive.ctr = character(0), negative.ctr = character(0),
  expwell = character(0),
  norm.method = getOption("opm.normalization.method"),
  QC.threshold = getOption("opm.QC.threshold"))

## S4 method for signature 'cellData,character,ANY'
x[i]

## S4 method for signature 'cellData'
show(object)
```

### Arguments

| | |
|---|---|
| `name` | character, the analyzed item |
| `positive.ctr` | a character vector, the positive control well IDs, e.g. c("E05", "E06") |
| `negative.ctr` | a character vector, the positive control well IDs, e.g. c("B05", "B06") |
| `expwell` | include all wells except control and neglect.well if NULL |
| `norm.method` | character the normalization method. |
| `QC.threshold` | numeric, the thresholds in the quality control. |
| `x` | a cellData object |
| `i` | a requested slot name |
| `object` | a cellData class |

### Value

a cellData object

## Slots

`name` character, one parameter in the Columbus system report.

`posctrwell` a character vector, the positive control well IDs, e.g. B05.

`negctrwell` a character vector, the negative control well IDs, e.g. B05.

`expwell` a character vector, the sample well IDs, e.g. C15.

`cellNum` matrix, cell numbers

`origin.data` a numeric matrix, the raw data matrix with rows the well IDs and columns the plate IDs.

`norm.data` a numeric matrix, the normalized data.

`qc.data` a numeric matrix, the data after quality control, with the rows are "barcode:wellID" and columns are the data of all replicated samples and their means, and if they have passed the quality control.

`norm.method` character the normalization method.

`QC.threshold` numeric, the thresholds in the quality control.

`plate.quality` a logical matrix, the quality data with the rows are the barcode and columns are the replicateIDs.

`plate.quality.data` a list of plate correlations and plate z' factors

`Sig` a list of the following components:

- `SigMat`:a logic matrix marking the high and low expressed hits
- `threshold`:the threshold of the high and low expressed hits
- `stats`:the numbers of the high and low expressed hits
- `pvalue`:the pvalue of each sample by t tests

## Methods

**Constructor** `cellData(name, positive.ctr = character(0), negative.ctr = character(0),expwell = characte`

**Show** signature(object="cellLoad"). Displays object content as text.

**Accessor** x[i]. `x`: a cellData object; `i`: character, a cellData slot name.

## Examples

```
oneCell <- cellData(name = "Average Intensity of Nuclei",
positive.ctr = c("H02", "J02", "L02"),
negative.ctr = c("C23", "E23", "G23"))
oneCell
oneCell["name"]
```

## cellLoad *Data importing*

### Description

Extracts data of a specific type in a list of `expData` objects to initialize a `cellData` object.

### Usage

```
cellLoad(object, lstPlates, ...)

## S4 method for signature 'cellData'
cellLoad(object, lstPlates, positive.ctr = NULL,
  negative.ctr = NULL, neglect.well = NULL, expwell = NULL)
```

### Arguments

`object` a cellData object

`lstPlates` a list of expData objects

`...` other parameters

`positive.ctr` a character vector, the positive control well IDs, e.g. c("E05", "E06")

`negative.ctr` a character vector, the positive control well IDs, e.g. c("B05", "B06")

`neglect.well` a character vector, the neglect wells. Accept regular expression, e.g. c("*02", "*23")

`expwell` include all wells except control and neglect.well if NULL

### Details

negative.ctr accept regular expression

### Value

a `cellData` object, with initialized slot `origin.data`

### Examples

```
data(platemap)
platemap$Path <- file.path(
system.file("Test", package = "OperaMate"), platemap$Path)
data(demoCell)
datapath <- file.path(system.file("Test", package = "OperaMate"), "Matrix")
lstPlates <- loadAll(cellformat = "Matrix", datapath = datapath)
oneCell <- cellLoad(oneCell, lstPlates, neglect.well = c("*02", "*23"))
str(oneCell["origin.data"])
```

## cellMean *Mean of two cellData objects*

### Description

Merges the intensities in nucleus and cytoplasm to their averages for signature detection.

### Usage

```
cellMean(cell1, cell2, name)

## S4 method for signature 'cellData,cellData,character'
cellMean(cell1, cell2, name)
```

### Arguments

| | |
|---|---|
| `cell1` | one cellData object |
| `cell2` | another celldata object |
| `name` | the name of mean cellData object |

### Value

the mean cellData object

### Examples

```
data(demoCell)
meanCell <- cellMean(oneCell, oneCell, "meanCell")
meanCell
```

## cellNorm *Data normalization*

### Description

Normalizes raw data based on different normalization methods.

### Usage

```
cellNorm(object, norm.method)

## S4 method for signature 'cellData'
cellNorm(object,
  norm.method = getOption("opm.normalization.method"))
```

### Arguments

| | |
|---|---|
| `object` | a cellData object |
| `norm.method` | getOption("opm.normalization.method") |

### Details

Method description: "MP" employes the median polish algorithm which divides data by the median of their plates and wells recursively, while "PMed" only divides data by the median of their plates; "Z" substracts data by their plate medians, and then divides by the median absolute deviations; "Ctr" divides data by the mean of their plate negative controls; "None" avoids the data normalization in this step. The first three methods are based on the assumption that most samples display no biological effects in the assay be analyzed. They are often more effective than "Ctr" method as to the high throughput screening.

### Value

a `celldata` object with initialized slot `norm.data`

### Examples

```
data(demoCell)
oneCell <- cellNorm(oneCell, norm.method = "MP")
str(oneCell["norm.data"])
```

---

## `cellNumLoad` *Load cell number*

---

### Description

Load cell number

### Usage

```
cellNumLoad(object, object.cellnum)

## S4 method for signature 'cellData,cellData'
cellNumLoad(object, object.cellnum)
```

### Arguments

| | |
|---|---|
| `object` | a cellData object |
| `object.cellnum` | a cellData object for cell numbers |

### Value

a `cellData` object, with initialized slot `cellNum`

## Examples

```
data(demoCell)
data(demoCellNum)
oneCell <- cellNumLoad(oneCell, oneCellNum)
str(oneCell["cellNum"])
```

---

# cellQC *Quality control*

---

## Description

Checks quality of all plates and then wells.

## Usage

```
cellQC(object, qcType = NULL, qc.threshold = NULL,
  replace.badPlateData = TRUE, plot = TRUE,
  outpath = getOption("opm.outpath"), ...)

## S4 method for signature 'cellData'
cellQC(object, qcType = getOption("opm.QC.type"),
  qc.threshold = getOption("opm.QC.threshold"),
  replace.badPlateData = getOption("opm.replace.badPlateData"), plot = TRUE,
  outpath = getOption("opm.outpath"), ...)
```

## Arguments

| | |
|---|---|
| `object` | a cellData object |
| `qcType` | the type of quality control |
| `qc.threshold` | quality control thresholds |
| `replace.badPlateData` | if TRUE, replace the values of bad plate by their replicates |
| `plot` | if TRUE, plot figures |
| `outpath` | directory of output figures, default: getOption("opm.outpath") |
| `...` | arguments for the graphic device |

## Details

Requires three or more replicated samples.

qcType include c("plateCorrelation", "wellSd", "zFactor", "cellNumber"), An example of qc.threshold is c(correlation = 0.8, zfactor = 0.5, cellnumber = 50).

## Value

a `cellData` object with intialized slot `qc.data`, `plate.quality` and `plate.quality.data`.

### Examples

```
data(demoCell)
op <- options("device")
options("device" = "png")
oneCell <- cellQC(oneCell, qcType = c("plateCorrelation", "wellSd", "cellNumber"),
qc.threshold = c(correlation = 0.7), outpath = tempdir())
options(op)
str(oneCell["qc.data"])
str(oneCell["plate.quality"])
```

## cellSig *Hit identification*

### Description

Detects samples those are most different from the negative controls.

### Usage

```
cellSig(object, method = c("stable", "ksd", "kmsd"), th = NULL,
  thPval = 0.05, digits = 3, adjust.method = p.adjust.methods,
  plot = TRUE, outpath = getOption("opm.outpath"), ...)

## S4 method for signature 'cellData'
cellSig(object, method = c("stable", "ksd", "kmsd"),
  th = NULL, thPval = 0.05, digits = 3,
  adjust.method = p.adjust.methods, plot = TRUE,
  outpath = getOption("opm.outpath"), ...)
```

### Arguments

| | |
|---|---|
| `object` | a cellData object |
| `method` | method = c("stable","ksd","kmsd"). Details are referred in the vignette. |
| `th` | numeric, the thresholds. It can be one threshold for both high and low expressed hit or two thresholds for each respectively. |
| `thPval` | numeric, threshold of pvalues in the t-test between the sample and control replicates |
| `digits` | integer, the number of digits used to show the thresholds |
| `adjust.method` | pvalue correction method |
| `plot` | plot QQ-plot when method is "stable" if TRUE. |
| `outpath` | directory of output figures, default: getOption("opm.outpath") |
| `...` | arguments of the graphic device |

### Value

a `cellData` object with initialized slot `Sig`.

### Examples

```
data(demoCell)
op <- options("device")
options("device" = "png")
oneCell <- cellSig(oneCell, method = "stable", th = c(0.05, 0.05),
outpath = tempdir())
options(op)
names(oneCell["Sig"])
```

---

## cellSigAnalysis *Hits function analysis*

---

### Description

Performs function analysis using gProfileR

### Usage

```
cellSigAnalysis(object, genemap, organism, type = c("High", "Low"),
  file = NULL, ...)
```

### Arguments

| | |
|---|---|
| `object` | a cellData object |
| `genemap` | a data frame, the well-gene specification table |
| `organism` | organism name. |
| `type` | include both high and low expressed hits or one of them. |
| `file` | the filename of the enrichment table (default: disabled) |
| `...` | the arguments of `gprofiler`. |

### Details

genemap must include colnames "Barcode","Well","GeneSymbol". organism name can be referred to g:Profiler tool. For example, human: hsapiens, mouse: mmusculus.

### Value

a data frame of the functional report from gProfiler

### Examples

```
data(demoCell)
genemap <- read.csv(file.path(system.file("Test", package = "OperaMate"),
"demoData", "genemap.csv"), stringsAsFactors = FALSE)
chart <- cellSigAnalysis(oneCell, genemap, organism = "mmusculus")
head(chart)
```

## `cellSigAnalysisPlot` *The barplot of enrichment functions*

### Description

The barplot of enrichment functions

### Usage

```
cellSigAnalysisPlot(chart, prefix = "", type = NULL, fill = "steelblue",
  outpath = getOption("opm.outpath"), ...)
```

### Arguments

| | |
|---|---|
| chart | data frame, the functional annotation chart |
| prefix | character, the prefix of figure name |
| type | selected domains from chart, e.g. BP. |
| fill | color of the bars |
| outpath | directory of output figures, default: getOption("opm.outpath") |
| ... | other arguments for graphical devices |

### Value

Invisibly the ggplot2 function for barplot

### Examples

```
data(demoCell)
genemap <- read.csv(file.path(system.file("Test", package = "OperaMate"),
"demoData", "genemap.csv"), stringsAsFactors = FALSE)
chart <- cellSigAnalysis(oneCell, genemap, organism = "mmusculus")
op <- options("device")
options("device" = "png")
cellSigAnalysisPlot(chart, type = "BP", outpath = tempdir())
options(op)
```

## cellSigPlot *Hits volcano plot*

### Description

Visualizes hits by volcano plot.

### Usage

```
cellSigPlot(object, outpath = getOption("opm.outpath"),
  color.highlight = getOption("opm.sig.color.highlight"),
  color.background = getOption("opm.sig.color.background"),
  highlight.label = NULL,
  highlight.label.color = getOption("opm.sig.label.color"), ...)
```

### Arguments

`object` a cellData object

`outpath` diretory of the output figures

`color.highlight` a character specifying the color of the hits

`color.background` a character specifying the color of the other samples

`highlight.label` a vector of characters specifying the names of the samples to be highlighted, with the names are the "barcode:wellID".

`highlight.label.color` a character specifying the color of the labels

`...` arguments of the graphic device and ggplot2

### Details

Users can highlight a certain samples during plotting.

### Value

Invisibly an object of `ggplot`

### Examples

```
data(demoCell)
op <- options("device")
options("device" = "png")
labels <- c("Axin1")
names(labels) <- c("DSIMGA04:C07")
cellSigPlot(oneCell, highlight.label = labels, outpath = tempdir())
options(op)
```

## cellViz *Data visualization*

### Description

Visualize data by heatmap or boxplot.

### Usage

```
cellViz(object, data.type = c("raw", "norm"), plot = c("heatmap",
  "boxplot"), outpath = getOption("opm.outpath"), multiplot = FALSE,
  plateID = NULL, tag = NULL, ctr.excluded = TRUE, ...)
```

### Arguments

| | |
|---|---|
| `object` | a cellData object |
| `data.type` | c("raw", "norm), visualizing both types by default |
| `plot` | c("heatmap","boxplot") |
| `outpath` | directory of output figures, default: getOption("opm.outpath") |
| `multiplot` | logical, the output images are placed in one figure or not |
| `plateID` | numeric or character |
| `tag` | character, unique tag for one figure |
| `ctr.excluded` | logical, if controls are included in the visualization |
| `...` | other arguments for graphical devices and `pheatmap` |

### Details

By visualizing the raw data, users can observe the batch effects as a large region of distinguishing color in heatmap or biased distribution by boxplots. Users can also visualize thr normalized data for comparison.

### Value

Invisibly a list of the values returned by `pheatmap` and ggplot2 function for boxplot

### Examples

```
data(demoCell)
op <- options("device")
options("device" = "png")
cellViz(oneCell, data.type = c("raw", "norm"), plateID = 1:6, outpath = tempdir())
cellViz(oneCell, data.type = c("raw", "norm"), plateID = 1, outpath = tempdir())
options(op)
```

## demoData *Examples of tables and cellData objects*

### Description

oneCellNum

### Value

platemap: a data frame

oneCell: a `cellData` object

oneCellNum: a `cellData` object

### platemap

**Description** The experiment information of each Columbus analysis report. This table is required only if the report formats are not standarded. See `loadAll` for more information.

**Format** data.frame with the following required column names:

- `FileName`**:** character, the name of the report.
- `Format`**:** character, only ”Tab” and ”Matrix” are supported in the current version.
- `Barcode`**:** character, the barcode of the plates.
- `RepID`**:** character, the ID to distinguish the replicated plates.
- `Path`**:** character, the full path of the report.

### oneCell

**Description** `oneCell` is a `cellData` object used in the examples of the package.

### oneCellNum

**Description** `oneCellNum` is a `cellData` object storing the cell numbers.

### Examples

```
data(platemap)
str(platemap)
data(demoCell)
oneCell
data(demoCellNum)
```

## expData-class *The expData class*

### Description

The expData class is a container to store data imported from one Columbus system report

Constructor method of expData class.

Show method

### Usage

```
expData(name, path, rep.id, exp.id, format)

## S4 method for signature 'expData'
show(object)

## S4 method for signature 'expData,ANY,ANY'
x[i]

dataLoad(object, data, wellID)

## S4 method for signature 'expData'
dataLoad(object, data, wellID)
```

### Arguments

| | |
|---|---|
| name | character, the plate ID (barcode-replicateID), e.g. DSIMGA03-s1. |
| path | character, the path of the Columbus system report. |
| rep.id | character, replicateID, e.g. s1. |
| exp.id | character, barcode, e.g. DSIMGA03. |
| format | character, format of the Columbus system report. |
| object | a expData class |
| x | a expData object |
| i | a requested slot name |
| data | the vectorized raw data matrix of one plate of each type. |
| wellID | a character vector, the well IDs. |

### Value

an expData object

### Slots

`name` character, the plate ID (barcode-replicateID), e.g. DSIMGA03-s1.

`path` character, the path of the Columbus system report.

`rep.id` character, replicateID, e.g. s1.

`exp.id` character, barcode, e.g. DSIMGA03.

`data` a list of vectors, the vectorized raw data matrix of one plate of each type.

`format` character, format of the Columbus system report.

`wellID` a character vector, the well IDs.

### Methods

**Constructor** `expData(name, path, rep.id, exp.id, format)`.

**Show** signature(object = "expData"). Displays object content as text.

**Accessor** x[i]. x: an expData object; i: character, an expData slot name.

**dataLoad** `dataLoad(object, data, wellID)`

### Examples

```
onePlate <- expData(name = "130504-s1-02.txt",
         path = file.path(system.file("Test", package = "OperaMate"),
                          "Matrix", "130504-s1-02.txt"),
       rep.id = "s1",
       exp.id = "DSIMGA02",
       format = "Matrix")
onePlate
onePlate["name"]
```

---

## `generateReport` *Report generation*

---

### Description

Summarizes all results in the list of `cellData` objects, and writes out a report to file.

### Usage

```
generateReport(lstCells, genemap = NULL, verbose = FALSE, file = NULL,
  outpath = getOption("opm.outpath"), plot = TRUE, ...)
```

### Arguments

| | |
|---|---|
| lstCells | a list of cellData objects |
| genemap | a data frame, the well-gene specification table |
| verbose | logical, detailed data will be provided if TRUE |
| file | the path of the file to generate to |
| outpath | a character string naming the location the figures to generate to |
| plot | if TRUE, plot barplot |
| ... | arguments of the graphic device |

### Details

This function summarizes the information from all cellData objects, and visualizes the number of the hists if required.

### Value

a data frame with annotated information of each well

### Examples

```
data(demoCell)
genemap <- read.csv(file.path(system.file("Test", package = "OperaMate"),
"demoData", "genemap.csv"), stringsAsFactors = FALSE)
report <- generateReport(list(oneCell), genemap, verbose = FALSE,
plot = FALSE)
str(report)
```

## loadAll *Data importing*

### Description

Initializes a list of expData objects from the Columbus system reports.

### Usage

```
loadAll(cellformat = NULL, datapath = "./",
  egFilename = getOption("opm.filename.example"), well.digits = 2,
  platemap = NULL)
```

## Arguments

`cellformat` character specifying the format of the reports. Enable when `platemap` is NULL.

`datapath` character specifying the location of the reports. Enable when `platemap` is NULL.

`egFilename` a file name example

`well.digits` the digits of the well column in the well-gene

`platemap` data frame. See an example as `platemap`.

## Details

To facility the automatic file name parsing, the reports obtained from Columbus system should be of the same format, and located under the same directory. Users can obtain this plate specification table for further modification. An example of the table can be referred by `platemap`. After modification, users can submit a plate speficication data frame to parameter `platemap`. The data format supported for the reports are "Tab" and "Matrix". If the reports are of other cellformats, you can specify its cellformat and rewrite the function `parseTemplete` to import the data seperately.

An example of egFilename = list(eg.filename = "0205-s2-01.txt", rep.id = "s2", exp.id = "01", sep = "-", barcode = "DSIMGA01"). well.digits: In the well-gene specification file, if the well ID is B1, B2, ..., B11, the well.digit = 1; while B01, B02, ..., B11, the well.digit = 2; and B001, B002, ..., B011, the well.digit =3.

## Value

a list of `expData` objects

## Examples

```
# Data frame \code{platemap} provided
data(platemap)
platemap$Path <- file.path(
system.file("Test", package = "OperaMate"), platemap$Path)
lstPlates <- loadAll(platemap = platemap)
#
# Consistent file name format
datapath <- file.path(system.file("Test", package = "OperaMate"), "Tab")
egFilename <- list(eg.filename = "Tab.130504-s1-01.txt",
rep.id = "s1", exp.id = "01", sep = "-",
barcode = "DSIMGA01")
lstPlates <- loadAll(cellformat = "Tab", datapath = datapath,
egFilename = egFilename, well.digits = 2)
#
lstPlates[[1]]
```

## nameParser *Plate information extraction*

### Description

Extract plate information from file names.

### Usage

```
nameParser(vec.files, egFilename)
```

### Arguments

| | |
|---|---|
| `vec.files` | a vector of file names |
| `egFilename` | a file name example |

### Details

An example of egFilename = list(eg.filename = "0205-s2-01.txt", rep.id = "s2", exp.id = "01", sep = "-", barcode = "DSIMGA01").

### Value

a data frame of PlateID, RepID, and Barcode

## operaMate *Data process and analysis pipeline*

### Description

A systematical pipeline for opera data importing, normalization, quality control, hit detection, analysis, and visualization.

### Usage

```
operaMate(configFile, gDevice = "png", ...)
```

### Arguments

| | |
|---|---|
| `configFile` | the location of the file specifying all parameters |
| `gDevice` | the graphics device |
| `...` | addition arguments for graphics devices |

### Value

a list of three components: a list of cellData objects, the annotated table of each well, and the enrichment analysis table

### Examples

```
configFile <- file.path(system.file("Test", package = "OperaMate"),
"demoData", "demoParam.txt")
operaReport <- operaMate(configFile, gDevice = "png")
head(operaReport$report)
```

## parseTemplete *Data extraction from one report*

### Description

Extracts data in the report to the slot `data` in the `expData` object. An inner function of `loadAll`.

### Usage

```
parseTemplete(onePlate, well.digits = 2)
```

### Arguments

| | |
|---|---|
| `onePlate` | an expData object |
| `well.digits` | the digits of the well column in the well-gene specification file |

### Value

an `expData` object with initialized slot `data`.

### Examples

```
datapath <- file.path(system.file("Test", package = "OperaMate"), "Tab")
lstPlates <- loadAll(cellformat = "Tab", datapath = datapath )
onePlate <- parseTemplete(lstPlates[[1]])
```

# Index