# Package: frapplot (via r-universe)

August 20, 2024

**Type** Package

**Title** Automatic Data Processing and Visualization for FRAP

**Version** 0.1.3

**Author** Guanqiao Ding <`gding16@gmail.com`>

**Maintainer** Guanqiao Ding <`gding16@gmail.com`>

**Description** Automatically process Fluorescence Recovery After Photobleaching (FRAP) data and generate consistent, publishable figures. Note: this package does not replace 'ImageJ' (or its equivalence) in raw image quantification. Some references about the methods: Sprague, Brian L. (2004) <doi:10.1529/biophysj.103.026765>; Day, Charles A. (2012) <doi:10.1002/0471142956.cy0219s62>.

**Depends** R (>= 2.10)

**Imports** grDevices, graphics, stats, utils

**BugReports** `https://github.com/GuanqiaoDing/frapplot/issues`

**URL** `https://github.com/GuanqiaoDing/frapplot`

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 6.1.1

**Repository** https://guanqiaoding.r-universe.dev

**RemoteUrl** https://github.com/guanqiaoding/frapplot

**RemoteRef** HEAD

**RemoteSha** 05c11015b052b77ac0bc2084c688b9bc250da6f9

# Contents

---

## `example_dataset` *Example dataset*

### Description

Example dataset

### Usage

```
example_dataset
```

### Format

A list of three matrices: each contains FRAP data for a control or experimental group. For each matrix, nrow = time_points + 1, ncol = sample size.

---

## `exclude` *Exclude samples from the dataset*

### Description

If certain samples are of poor quality, use this function to exclude them from the dataset.

### Usage

```
exclude(ds, group, cols)
```

### Arguments

| | |
|---|---|
| `ds` | Name of the dataset. |
| `group` | Name of the group from which to exclude certain samples. |
| `cols` | A vector of numbers specifying the column(s) to exclude. |

### Value

Modified dataset in the same format.

## Examples

```
ds <- exclude(example_dataset, group = "mut1", cols = c(1,3))
```

---

# `frapplot` *Plot FRAP data of two selected groups*

## Description

Plot FRAP data of any two groups (e.g. control and mutant) in a consistent and publishable format.

## Usage

```
frapplot(path, control, mutant, info)
```

## Arguments

| | |
|---|---|
| `path` | Path of the output directory |
| `control` | Name of the control. |
| `mutant` | Name of the mutant. |
| `info` | Returned information from `frapprocess()`. |

## Examples

```
info <- frapprocess(example_dataset, seq(0, 145, 5))
frapplot(tempdir(), "control", "mut2", info)
```

---

# `frapprocess` *Process FRAP data*

## Description

Normalize and analyze FRAP data. Perform non-linear regression and calculate ymax, ymin, k, halftime, tau, total_recovery, total_recovery_sd.

## Usage

```
frapprocess(ds, time_points)
```

## Arguments

| | |
|---|---|
| `ds` | A dataset that contains FRAP data for multiple experiment groups |
| `time_points` | A vector of time points (in second) that the experiment uses, e.g. 0, 5, 10, .... |

## Value

A list of results:

- $time_points: a vector of time points
- $summary: summary of the regression
- $sample_means: a matrix of sample means, nrow = num of time points, ncol = sample size
- $sample_sd: a matrix of standard deviations, nrow = num of time points, ncol = sample size
- $model: a list of models for each group from the non-linear regression
- $details: details of the regression for each group

## Examples

```
info <- frapprocess(example_dataset, seq(0, 145, 5))
```

# Index