

Package ‘imbibe’

May 8, 2026

Version 0.1.1

Date 2022-11-04

Title A Pipe-Friendly Image Calculator

Maintainer Jon Clayden <code@clayden.org>

Imports Rcpp, RNifti, magrittr

LinkingTo Rcpp, RNifti

Suggests mmand, tinytest, covr

Description Provides a set of fast, chainable image-processing operations which are applicable to images of two, three or four dimensions, particularly medical images.

License BSD_3_clause + file LICENCE

URL <https://github.com/jonclayden/imbibe>

BugReports <https://github.com/jonclayden/imbibe/issues>

Encoding UTF-8

RoxygenNote 7.2.1

NeedsCompilation yes

Author Jon Clayden [aut, cre] (ORCID: <<https://orcid.org/0000-0002-6608-0619>>),
Chris Rorden [aut] (ORCID: <<https://orcid.org/0000-0002-7554-6142>>),
John Muschelli [ctb] (ORCID: <<https://orcid.org/0000-0001-6469-1750>>),
Exstrom Laboratories LLC [cph]

Repository CRAN

Date/Publication 2022-11-09 21:50:16 UTC

Contents

add	2
dilate	3
dim_mean	4
expect_pipeline_result	4
exponent	5

imbibe	6
kernel_3d	7
run	8
threshold	8

Index	10
--------------	-----------

add	<i>Basic binary operations</i>
-----	--------------------------------

Description

Basic binary operations

Usage

add(image, arg)

subtract(image, arg)

multiply(image, arg)

divide(image, arg)

remainder(image, arg)

mask(image, arg)

maximum(image, arg)

minimum(image, arg)

Arguments

image An image object or pipeline.

arg Numeric or image argument.

Value

An updated pipeline.

dilate *Mathematical morphology and filtering operations*

Description

Mathematical morphology and filtering operations

Usage

```
dilate(image, kernel = NULL, ..., max = FALSE, nonzero = TRUE)
```

```
dilateall(image, kernel = NULL, ...)
```

```
erode(image, kernel = NULL, ..., min = FALSE)
```

```
filter_median(image, kernel = NULL, ...)
```

```
filter_mean(image, kernel = NULL, ..., norm = TRUE)
```

```
smooth_gauss(image, sigma)
```

```
subsample(image, offset = FALSE)
```

Arguments

image	An image object or pipeline.
kernel	A suitable kernel function (see kernels). If NULL, the most recently set kernel in the pipeline is used, if any, otherwise the default kernel (kernel_3d).
...	Additional arguments to the kernel function, if any.
max	Logical value: if TRUE, maximum filtering is used for dilation; otherwise mean filtering is used. Mean filtering is always used by dilateall.
nonzero	Logical value: if TRUE, the default, dilation is only applied to nonzero pixels/voxels. Otherwise it is applied everywhere (and maximum filtering is always used).
min	Logical value: if TRUE, minimum filtering is used for erosion; otherwise nonzero voxels overlapping with the kernel are simply zeroed.
norm	Logical value indicating whether the mean filter will be normalised or not.
sigma	Numeric value giving the standard deviation of the Gaussian smoothing kernel.
offset	Logical value indicating whether subsampled pixels should be offset from the original locations or not.

Value

An updated pipeline.

dim_mean	<i>Dimensionality reduction operations</i>
----------	--

Description

Dimensionality reduction operations

Usage

```
dim_mean(image, dim = 4L)
dim_sd(image, dim = 4L)
dim_max(image, dim = 4L)
dim_whichmax(image, dim = 4L)
dim_min(image, dim = 4L)
dim_median(image, dim = 4L)
dim_quantile(image, dim = 4L, prob)
dim_AR1(image, dim = 4L)
```

Arguments

image	An image object or pipeline.
dim	Integer value between 1 and 4, giving the dimension to apply the reduction along.
prob	For <code>drt_quantile</code> , the quantile probability to extract (analogously to quantile).

Value

An updated pipeline.

expect_pipeline_result	<i>Expectation for testing pipeline output</i>
------------------------	--

Description

This function provides an expectation for use with the "tinytest" package, which runs the pipeline specified in its first argument and compares the result to its second.

Usage

```
expect_pipeline_result(current, target, precision = "double", ...)
```

Arguments

current	The pipeline to run, which should have class "imbibe".
target	The target value to compare against, a numeric array of some kind, which will be converted to a "niftiImage" object.
precision	A string specifying the working precision. Passed to <code>run</code> .
...	Further arguments to <code>expect_equal</code> .

Value

A "tinytest" object.

exponent	<i>Basic unary operations</i>
----------	-------------------------------

Description

Basic unary operations

Usage

```
exponent(image)
```

```
logarithm(image)
```

```
sine(image)
```

```
cosine(image)
```

```
tangent(image)
```

```
arcsine(image)
```

```
arccosine(image)
```

```
arctangent(image)
```

```
square(image)
```

```
squareroot(image)
```

```
reciprocal(image)
```

```
absolute(image)

binarise(image, invert = FALSE)

binarize(image, invert = FALSE)
```

Arguments

image	An image object or pipeline.
invert	Logical value: if TRUE, binarising will also perform logical inversion so that only zeroes in the original image will be nonzero; if FALSE, the default, the usual sense is used, in which zeroes remain as they are, and everything else is converted to 1.

Value

An updated pipeline.

imbibe	<i>Create an operation pipeline</i>
--------	-------------------------------------

Description

Create an operation pipeline

Usage

```
imbibe(image)

## S3 method for class 'imbibe'
asNifti(x, ...)

## S3 method for class 'imbibe'
as.array(x, ...)

## S3 method for class 'imbibe'
print(x, ...)
```

Arguments

image	An image object or existing pipeline.
x	An "imbibe" object.
...	Additional arguments to methods.

kernel_3d	<i>Mathematical morphology kernels</i>
-----------	--

Description

Mathematical morphology kernels

Usage

```
kernel_3d(image)
```

```
kernel_2d(image)
```

```
kernel_box(image, width, voxels = FALSE)
```

```
kernel_gauss(image, sigma)
```

```
kernel_sphere(image, radius)
```

```
kernel_file(image, file)
```

Arguments

image	An image object or pipeline.
width	The width of the kernel in appropriate units. If <code>voxels</code> is <code>FALSE</code> a value can be specified for each of the three dimensions; otherwise only a single value should be given and the kernel will be isotropic.
voxels	Logical value: if <code>TRUE</code> , the width is given in pixels/voxels and must be an odd integer; otherwise, the units are millimetres and can take any value.
sigma	Numeric value giving the standard deviation of a Gaussian kernel, in millimetres.
radius	Numeric value giving the radius of a sphere kernel, in millimetres.
file	Name of a NIFTI file containing the kernel.

Value

An updated pipeline.

run	<i>Run a pipeline and return an image result</i>
-----	--

Description

Run a pipeline and return an image result

Usage

```
run(pipe, precision = getOption("imbibe.precision", "double"))
```

Arguments

pipe	An operation pipeline.
precision	The internal precision used for calculations. May be "double", "float" or "single"; the latter two are equivalent.

Value

An image

Examples

```
im <- RNifti::readNifti(system.file("extdata", "example.nii.gz", package="RNifti"))
pipe <- im %>% threshold_below(500) %>% binarise()
run(pipe)
```

threshold	<i>Image thresholding</i>
-----------	---------------------------

Description

Image thresholding

Usage

```
threshold(
  image,
  value,
  reference = c("none", "image", "nonzero"),
  above = FALSE
)

threshold_below(image, value, reference = c("none", "image", "nonzero"))

threshold_above(image, value, reference = c("none", "image", "nonzero"))
```

Arguments

image	An image object or pipeline.
value	Numeric threshold value.
reference	String indicating what the value should be referenced against, if anything. If "none", the default, the value is taken literally. If "image", it is interpreted as a proportion of the "robust range" of the current image's intensities. If "nonzero" it is interpreted as a proportion of the "robust range" of the nonzero pixel intensities.
above	Logical value: if TRUE the operation zeroes values above the threshold; otherwise it zeroes values below it. The <code>threshold_below</code> and <code>threshold_above</code> function variants set argument implicitly.

Value

An updated pipeline.

Index

absolute (exponent), 5
add, 2
arccosine (exponent), 5
arcsine (exponent), 5
arctangent (exponent), 5
as.array.imbibe (imbibe), 6
asNifti.imbibe (imbibe), 6

binarise (exponent), 5
binarize (exponent), 5

cosine (exponent), 5

dilate, 3
dilateall (dilate), 3
dim_AR1 (dim_mean), 4
dim_max (dim_mean), 4
dim_mean, 4
dim_median (dim_mean), 4
dim_min (dim_mean), 4
dim_quantile (dim_mean), 4
dim_sd (dim_mean), 4
dim_whichmax (dim_mean), 4
divide (add), 2

erode (dilate), 3
expect_pipeline_result, 4
exponent, 5

filter_mean (dilate), 3
filter_median (dilate), 3

imbibe, 6

kernel_2d (kernel_3d), 7
kernel_3d, 7
kernel_box (kernel_3d), 7
kernel_file (kernel_3d), 7
kernel_gauss (kernel_3d), 7
kernel_sphere (kernel_3d), 7
kernels, 3

kernels (kernel_3d), 7

logarithm (exponent), 5

mask (add), 2
maximum (add), 2
minimum (add), 2
multiply (add), 2

print.imbibe (imbibe), 6

quantile, 4

reciprocal (exponent), 5
remainder (add), 2
run, 5, 8

sine (exponent), 5
smooth_gauss (dilate), 3
square (exponent), 5
squareroot (exponent), 5
subsample (dilate), 3
subtract (add), 2

tangent (exponent), 5
threshold, 8
threshold_above (threshold), 8
threshold_below (threshold), 8