Package: bgnormR (via r-universe)

November 10, 2026

Type Package

Title Background Normalisation for Multiplex Spatial Proteomics

Version 0.99.5

Description Implements bgnorm, a generative statistical framework for background correction, normalisation, and quality control in multiplex spatial proteomics. The method models fluorescence pixel intensities as a three-component mixture of log-normal convolutions representing background, non-specific binding, and biological signal. A deconvolution step separates the signal component from background sources, and a model-based quantile normalisation (bgnormQ) unifies measurements across markers, samples, and three-dimensional tissue slices. A Jensen-Shannon Divergence metric provides automated quality control. A Java-free reader for Akoya PhenoCycler-Fusion QPTIFF, OME-TIFF, and OME-Zarr (OME-NGFF) images is included.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.5.0)

Imports mclust (>= 5.4), SummarizedExperiment (>= 1.28.0), SpatialExperiment (>= 1.10.0), BiocParallel (>= 1.30.0), DelayedArray (>= 0.26.0), S4Vectors, ggplot2 (>= 3.4.0), scales, stats, utils, methods

Suggests SingleCellExperiment (>= 1.20.0), scattermore, tiff (>= 0.1-11), xml2 (>= 1.3), jsonlite, ijtiff, Rarr, BiocStyle, knitr, rmarkdown, testthat (>= 3.0.0), EBImage

VignetteBuilder knitr

biocViews Normalization, Preprocessing, SingleCell, Spatial, Proteomics, ImmunoOncology, QualityControl, DataImport

Config/testthat/edition 3

URL https://github.com/BhuvaLab/bgnormR

BugReports https://github.com/BhuvaLab/bgnormR/issues

Config/roxygen2/version 8.0.0

Config/pak/sysreqs libmagick++-dev gsfonts libicu-dev libssl-dev zlib1g-dev

Repository https://biocstaging.r-universe.dev

Date/Publication 2026-11-08 11:09:56 UTC

RemoteUrl https://github.com/BiocStaging/bgnormR

RemoteRef HEAD

RemoteSha b187a5be699d2bb126555d0af1734c6852d6adf4

Contents

bgnormR-package . . . 3
[.QPTIFFImage . . . 4
as.array.QPTIFFImage . . . 5
as.list.QPTIFFImage . . . 5
as.matrix.QPTIFFImage . . . 6
as.QPTIFFImage . . . 6
bgnorm_cells . . . 7
bgnorm_pixels . . . 8
bgnorm_results . . . 10
bgnorm_sce . . . 11
BgnormResult . . . 12
channel_table . . . 13
dimnames<-.QPTIFFImage . . . 13
inv_log_transform . . . 14
jsd_gaussians . . . 15
jsd_qc . . . 15
length.QPTIFFImage . . . 16
log_transform . . . 16
median_filter_3x3 . . . 17
metadata . . . 18
names.QPTIFFImage . . . 19
names<-.QPTIFFImage . . . 19
OMEZarrArraySeed-class . . . 20
plot_distributions . . . 21
plot_jsd_heatmap . . . 22
plot_pixel_classes . . . 23
plot_qptiff . . . 24
qc_summary . . . 26
qpi_channel_names . . . 27
qpi_channels . . . 27
qpi_format . . . 28
qpi_is_brightfield . . . 28
qpi_n_levels . . . 29
qpi_pixel_size_um . . . 29
qpi_primary_scene . . . 30
QPTIFFArraySeed-class . . . 30

QPTIFFImage 32
QPTIFFMetadata 32
read_qptiff 32
write_qptiff 34

Index **36**

---

bgnormR-package *bgnormR: Background Normalisation for Multiplex Spatial Proteomics*

Description

Implements *bgnorm*, a generative statistical framework for background correction, normalisation, and quality control in multiplex spatial proteomics (Kharbanda and Tubelleza et al. 2026).

Details

The framework models log-transformed fluorescence intensities as a three-component Gaussian mixture representing:

1. Background (empty space / instrument noise)
2. Non-specific binding and autofluorescence
3. True biological signal

Core functions:

- `bgnorm_pixels` - pixel-level background correction
- `bgnorm_cells` - cell-level background correction
- `bgnorm_sce` - normalise a SummarizedExperiment / SpatialExperiment
- `jsd_qc` - Jensen-Shannon Divergence quality metric
- `read_qptiff` - Akoya PhenoCycler-Fusion image reader
- `plot_qptiff`, `plot_pixel_classes`, `plot_distributions`, `plot_jsd_heatmap` - plotting utilities

Author(s)

Maintainer: Dharmesh D. Bhuva <d.bhuva@uq.edu.au> (ORCID)

Authors:

- Dharmesh D. Bhuva <d.bhuva@uq.edu.au> (ORCID)
- Rafael Tubelleza <r.tubelleza@uq.edu.au> (ORCID)
- Malvika Kharbanda <malvika.kharbanda@adelaide.edu.au> (ORCID)
- Agus Salim <salim.a@unimelb.edu.au> (ORCID)

See Also

Useful links:

- https://github.com/BhuvaLab/bgnormR
- Report bugs at https://github.com/BhuvaLab/bgnormR/issues

`[.QPTIFFImage` *Subset a QPTIFFImage*

Description

Standard 3-D array subscripting with all missing-subscript combinations supported. The result is *always* a `QPTIFFImage`, including when a single channel is selected. To extract a plain 2-D matrix for one channel, use `unclass(img[, , "DAPI"])[, , 1L]`.

- `img[, , "DAPI"]` - single channel (returns a 1-channel QPTIFFImage)
- `img[, , c("DAPI","CD3")]` - multi-channel sub-image
- `img[1:512, 1:512, ]` - spatial crop

Usage

```
## S3 method for class 'QPTIFFImage'
x[i, j, k, drop = FALSE]
```

Arguments

`x`	A `QPTIFFImage`.
`i`	Row indices (spatial), or missing.
`j`	Column indices (spatial), or missing.
`k`	Channel indices (numeric or character), or missing.
`drop`	Ignored; the result is always a 3-D `QPTIFFImage`.

Value

A `QPTIFFImage`.

as.array.QPTIFFImage *Materialise a QPTIFFImage to a plain 3-D array*

Description

For eager QPTIFFImages, equivalent to `unclass(x)`. For lazy (on-disk) QPTIFFImages, reads all pixel data from disk.

Usage

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

Arguments

`x`	A `QPTIFFImage`.
`...`	Unused.

Value

A plain numeric array of dimension `c(H, W, C)`.

as.list.QPTIFFImage *Convert a QPTIFFImage to a named list of 2-D channel matrices*

Description

Convert a QPTIFFImage to a named list of 2-D channel matrices

Usage

```
## S3 method for class 'QPTIFFImage'
as.list(x, ...)
```

Arguments

`x`	A `QPTIFFImage`.
`...`	Unused.

Value

A named list of 2-D matrices, one per channel.

as.matrix.QPTIFFImage *Coerce a single-channel QPTIFFImage to a matrix*

Description

Drops the singleton channel dimension and returns a plain `H x W` matrix. The channel name is stored in `attr(result, "channel")`. If the object has more than one channel, use `img[, , "DAPI"]` to select one first.

Usage

```
## S3 method for class 'QPTIFFImage'
as.matrix(x, ...)
```

Arguments

`x`	A `QPTIFFImage` with exactly one channel, *or* a 2-D array (already a matrix-like object), which is returned with `as.matrix` applied directly.
`...`	Unused.

Value

A numeric matrix of dimension `c(H, W)`.

Examples

```
arr <- array(1:12, dim = c(3, 4, 1),
             dimnames = list(NULL, NULL, "DAPI"))
img <- as.QPTIFFImage(arr)
m   <- as.matrix(img)
dim(m)          # 3 x 4
attr(m, "channel")  # "DAPI"
```

as.QPTIFFImage *Coerce a 3-D array to a QPTIFFImage*

Description

Attaches the `QPTIFFImage` class to any 3-D numeric array `[H x W x C]`. Channel names are taken from `dimnames(x)[[3]]`. This is the intended way to promote a plain array produced outside `read_qptiff` (e.g., the adjusted output of `bgnorm_markers`) into a first-class `QPTIFFImage`.

Usage

```
as.QPTIFFImage(x, ...)

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

## S3 method for class 'array'
as.QPTIFFImage(x, metadata = list(), ...)

## S3 method for class 'DelayedArray'
as.QPTIFFImage(x, metadata = list(), ...)
```

Arguments

x	A 3-D numeric array [H x W x C]. For a QPTIFFImage, the object is returned unchanged.
...	Unused; present for S3 method consistency.
metadata	Optional named list of metadata to embed. Ignored when x is already a QPTIFFImage.

Value

A QPTIFFImage.

Examples

```
arr <- array(runif(20 * 20 * 3), dim = c(20, 20, 3),
             dimnames = list(NULL, NULL, c("DAPI", "CD3", "CD8")))
img <- as.QPTIFFImage(arr)
class(img)
```

---

bgnorm_cells *Cell-level background normalisation (bgnorm)*

Description

Applies a two-component Gaussian Mixture Model to cell-level aggregated intensities for a single marker. The two components represent non-specific binding / autofluorescence and biological signal.

Usage

```
bgnorm_cells(
  x,
  cofactor = 150,
  quantile_norm = FALSE,
  quantile = 0.75,
  sample_prop = 1,
  ...
)
```

Arguments

x	Numeric vector of raw cell-level intensities for one marker. Must be a plain vector; matrices and arrays are not accepted.
cofactor	Positive numeric cofactor for log2 transform (default 150).
quantile_norm	Logical; apply model-based quantile normalisation?
quantile	Quantile of the signal component for normalisation (default 0.75).
sample_prop	Numeric in (0, 1]; proportion of non-zero observations used to fit the GMM. Default 1 (use all).
...	Additional arguments forwarded to the internal GMM fitter.

Details

This is a cell-level approximation of the pixel-level method intended for cases where only cell-summarised intensities (e.g., mean or median intensity per cell) are available.

Value

A BgnormResult object.

See Also

bgnorm_pixels, bgnorm_sce

Examples

```
set.seed(3)
# Simulate cell intensities: nonspecific + signal
x <- c(exp(rnorm(300, log(200), 0.5)), exp(rnorm(100, log(1500), 0.8)))
res <- bgnorm_cells(x)
print(res)
```

---

bgnorm_pixels *Pixel-level background normalisation (bgnorm) for a QPTIFFImage*

Description

Fits a three-component Gaussian Mixture Model (GMM) to the log2-transformed pixel intensities for each channel in a QPTIFFImage and deconvolves the biological signal component from background sources. All channels are processed independently; use channels to restrict processing to a subset. Optionally runs channels in parallel via **BiocParallel**.

Usage

```
bgnorm_pixels(
  img,
  channels = NULL,
  cofactor = 150,
  quantile_norm = FALSE,
  quantile = 0.75,
  sample_prop = 0.1,
  BPPARAM = BiocParallel::SerialParam(),
  ...
)
```

Arguments

img	A QPTIFFImage [H x W x C]. Use as.QPTIFFImage to promote a plain 3-D array or 2-D matrix.
channels	Character or integer vector of channels to process, or NULL (default) to process all channels.
cofactor	Positive numeric cofactor for the log2 transform. Default 150 is appropriate for 16-bit Akoya PhenoCycler-Fusion images.
quantile_norm	Logical; apply model-based quantile normalisation (bgnormQ)? Default FALSE.
quantile	Quantile of the signal component used for normalisation when quantile_norm = TRUE. Default 0.75.
sample_prop	Numeric in (0, 1]; proportion of non-zero pixels used to fit the GMM. Values < 1 draw a random subset, accelerating fitting on large images. Default 1 (all non-zero pixels).
BPPARAM	A BiocParallelParam instance. Default BiocParallel::SerialParam().
...	Additional arguments forwarded to the internal GMM fitter.

Details

The preprocessing, model, and deconvolution steps follow Kharbanda and Tubelleza et al. 2026:

1. Log2-transform: $I_{\log} = \log_2(I/c + 1)$.
2. Fit a 3-component GMM via EM (mclust), filtering zeros and optionally subsampling for speed.
3. Resolve component identities (background, non-specific, signal).
4. Apply the deconvolution formula to recover U_3.
5. Optionally normalise by the 75th-percentile of the signal component (bgnormQ).

Value

A QPTIFFImage [H x W x C] of background-adjusted log-intensities. The per-channel BgnormResult objects (GMM parameters, JSD) are stored as an attribute; retrieve them with bgnorm_results(result).

See Also

bgnorm_cells, bgnorm_results, jsd_qc, as.QPTIFFImage

Examples

```
set.seed(42)
px  <- c(exp(rnorm(500, log(50),   0.4)),
         exp(rnorm(400, log(300),  0.6)),
         exp(rnorm(100, log(2000), 1.0)))
arr <- array(c(px, px), dim = c(length(px), 1L, 2L),
             dimnames = list(NULL, NULL, c("DAPI", "CD3")))
img <- as.QPTIFFImage(arr)
res <- bgnorm_pixels(img)
names(res)                        # channel names of the adjusted image
bgnorm_results(res)[["DAPI"]]     # BgnormResult for the DAPI channel
# Process only selected channels
res2 <- bgnorm_pixels(img, channels = "DAPI")


# Real tissue image shipped with the package
path <- system.file("extdata", "PA_HNC_sample.ome.tiff", package = "bgnormR")
img  <- read_qptiff(path)
res  <- bgnorm_pixels(img, sample_prop = 0.1)
bgnorm_results(res)[["PanCK"]]    # per-channel model summary
```

---

bgnorm_results *Per-channel bgnorm results stored in a QPTIFFImage*

Description

Returns the named list of BgnormResult objects attached to a QPTIFFImage by bgnorm_pixels, or NULL if the image has not been background-normalised.

Usage

```
bgnorm_results(x, ...)
```

Arguments

x	A QPTIFFImage.
...	Unused.

Value

A named list of BgnormResult objects (one per processed channel), or NULL.

See Also

bgnorm_pixels

Examples

```
path <- system.file("extdata", "PA_HNC_sample.ome.tiff", package = "bgnormR")
img <- read_qptiff(path)
adj <- bgnorm_pixels(img, markers = names(img)[1])
bgnorm_results(adj)
```

---

bgnorm_sce *Apply cell-level bgnorm to a SingleCellExperiment or matrix*

Description

Runs bgnorm_cells independently for each marker (column) and returns an updated object with adjusted intensities in a new assay.

Usage

```
bgnorm_sce(
  x,
  assay.type = "counts",
  name = "bgnorm",
  cofactor = 150,
  quantile_norm = FALSE,
  quantile = 0.75,
  BPPARAM = BiocParallel::SerialParam(),
  ...
)
```

Arguments

x	A SummarizedExperiment (including SingleCellExperiment and SpatialExperiment subclasses) or a numeric matrix (cells × markers).
assay.type	Character; name of the assay to normalise. Default "counts".
name	Character; name for the output assay (SCE) or returned matrix attribute. Default "bgnorm".
cofactor	Cofactor for log2 transform (default 150).
quantile_norm	Logical; apply bgnormQ?
quantile	Quantile for normalisation.
BPPARAM	A BiocParallelParam instance.
...	Additional arguments forwarded to bgnorm_cells.

Value

The input object with a new assay (`name`) holding the adjusted intensities. The per-marker `BgnormResult` list is stored in `metadata(x)$bgnorm_results`.

Examples

```
library(SummarizedExperiment)
set.seed(5)
counts <- matrix(
  exp(rnorm(2000, log(300), 0.9)),
  nrow = 10, ncol = 200,
  dimnames = list(paste0("marker", seq_len(10)), paste0("cell", seq_len(200)))
)
se <- SummarizedExperiment(assays = list(counts = counts))
se <- bgnorm_sce(se, assay.type = "counts", name = "bgnorm")
assayNames(se)
```

`BgnormResult` *BgnormResult S3 class*

Description

A lightweight list-based S3 class returned by `bgnorm_pixels` (per channel, via `bgnorm_results`) and `bgnorm_cells`.

Value

Not applicable; documents the `BgnormResult` class structure.

Fields

`parameters` List with elements `means`, `sds`, `props`.

`n` Integer; total number of input observations (pixels or cells, including zeros) for this channel.

`threshold` For the 3-component (pixel-level) model: a single numeric scalar giving the maximum adjusted intensity of Non-specific pixels. Combined with the implicit class-1 boundary at 0, this encodes the full classification: `adj` == `0` -> Background; `0` < `adj` <= `threshold` -> Non-specific; `adj` > `threshold` -> Signal. `NULL` for 2-component models (cell-level or no-signal fallback), where the only boundary is 0 (adjusted > 0 -> Signal).

`jsd` Jensen-Shannon Divergence QC metric between components 2 and 3. `NA_real_` for no-signal channels.

`level` `"pixel"` or `"cell"`.

`quantile_norm` Logical; whether quantile normalisation was applied.

`histogram` List with `$breaks` and `$density` vectors from a pre-computed histogram of log2-transformed intensities. Used by `plot_distributions`.

`no_signal` Logical; `TRUE` when the 3-component GMM failed and the channel was fitted with a 2-component fallback, indicating no detectable biological signal.

bic Named numeric vector c(G2 = ..., G3 = ...) with mclust BIC values (higher is better in mclust's convention) divided by the number of points used to fit, giving a point-specific estimate comparable across channels regardless of sample size. NULL for cell-level results.

---

channel_table *Tidy per-channel metadata table*

Description

Collapses the primary scene's channel list into a one-row-per-channel data.frame for quick inspection. Multi-valued fields such as color_rgb are rendered as strings.

Usage

```
channel_table(meta)
```

Arguments

meta	A QPTIFFMetadata object.

Value

A data.frame with one row per channel; zero rows when the image has no channel metadata.

Examples

```
path <- system.file("extdata", "PA_HNC_sample.ome.tiff", package = "bgnormR")
img  <- read_qptiff(path)
channel_table(metadata(img))
```

---

dimnames<-.QPTIFFImage *Replace the dimnames of a QPTIFFImage*

Description

Low-level setter used by names<-.QPTIFFImage and direct dimnames(img) <- assignments. Preserves the QPTIFFImage class and the "metadata" attribute.

Usage

```
## S3 replacement method for class 'QPTIFFImage'
dimnames(x) <- value
```

Arguments

x	A QPTIFFImage.
value	A list of length 3 (NULL elements are allowed for the row and column dimensions).

Value

x with updated dimnames.

---

inv_log_transform *Inverse log2 transform with cofactor*

Description

Reverses log_transform: $I = (2^{I_{log}} - 1) \times$ cofactor.

Usage

```
inv_log_transform(x, cofactor = 150)
```

Arguments

x	Numeric vector or matrix of log2-transformed values.
cofactor	Positive numeric cofactor used during log_transform.

Value

Numeric vector or matrix on the original intensity scale.

Examples

```
x <- c(0, 100, 500, 2000, 65535)
x_log <- log_transform(x)
all.equal(x, inv_log_transform(x_log))
```

jsd_gaussians *Jensen-Shannon Divergence between two Gaussian distributions*

Description

Computes JSD numerically on a dense grid, used as a signal-to-noise QC metric between the non-specific binding component (2) and the biological signal component (3) of the bgnorm model. Higher values indicate better separation and staining quality.

Usage

```
jsd_gaussians(mu1, sd1, mu2, sd2, n_grid = 2000L)
```

Arguments

mu1, mu2	Means of the two Gaussian distributions.
sd1, sd2	Standard deviations of the two Gaussian distributions.
n_grid	Number of evaluation points (default 2000).

Value

Scalar JSD value in [0, 1] (bits, log2 base).

Examples

```
# Well-separated distributions -> high JSD
jsd_gaussians(mu1 = 0, sd1 = 0.5, mu2 = 3, sd2 = 0.8)
# Identical distributions -> JSD = 0
jsd_gaussians(mu1 = 1, sd1 = 1, mu2 = 1, sd2 = 1)
```

jsd_qc *Compute the JSD quality control metric from bgnorm model parameters*

Description

Extracts the JSD between components 2 (non-specific binding) and 3 (signal) from the fitted model parameters stored in a BgnormResult object.

Usage

```
jsd_qc(result)
```

Arguments

result	A BgnormResult object returned by bgnorm_pixels or bgnorm_cells.

Value

Named numeric scalar `jsd`.

Examples

```
set.seed(1)
x   <- c(exp(rnorm(300, log(300), 0.5)), exp(rnorm(100, log(2000), 0.9)))
res <- bgnorm_cells(x)
jsd_qc(res)
```

---

length.QPTIFFImage *Number of channels in a QPTIFFImage*

Description

Number of channels in a QPTIFFImage

Usage

```
## S3 method for class 'QPTIFFImage'
length(x)
```

Arguments

`x`	A `QPTIFFImage`.

Value

Integer; the number of channels.

---

log_transform *Log2 transform with cofactor*

Description

Applies $I_{log} = \log_2(I/\text{cofactor} + 1)$ to stabilise variance in low-intensity fluorescence signals. The default cofactor of 150 is appropriate for 16-bit images from the Akoya PhenoCycler-Fusion; adjust for other platforms.

Usage

```
log_transform(x, cofactor = 150)
```

Arguments

`x`	Numeric vector or matrix of raw intensities.
`cofactor`	Positive numeric cofactor (default 150 for 16-bit images).

Value

Numeric vector or matrix of log2-transformed values.

See Also

inv_log_transform

Examples

```
x <- c(0, 100, 500, 2000, 10000, 65535)
log_transform(x)
```

`median_filter_3x3` *Apply a 3 x 3 median filter to a matrix*

Description

Uses `EBImage::medianFilter` (C-based, fast) when **EBImage** is available, otherwise falls back to a pure-R sliding-window implementation.

Usage

```
median_filter_3x3(img)
```

Arguments

`img`	Numeric matrix (height x width).

Value

Filtered matrix of the same dimensions.

Examples

```
m <- matrix(runif(100), 10, 10)
m_filt <- median_filter_3x3(m)
```

`metadata` *Access the metadata embedded in a QPTIFFImage*

Description

Retrieves the `QPTIFFMetadata` object (an OME-organised hierarchy of slide / image / channel / scale metadata) stored inside a `QPTIFFImage`. The generic falls back to `metadata` for S4 objects such as `SummarizedExperiment`, so `metadata(img)` and `metadata(spe)` both work when bgnormR is loaded.

Usage

```
metadata(x, ...)

## S3 method for class 'QPTIFFImage'
metadata(x, ...)
```

Arguments

`x`	A `QPTIFFImage`.
`...`	Unused.

Details

The returned object mirrors the OME hierarchy:

`slide` Slide-constant identity (id, barcode, operator, ...).

`images` One scene per distinct image, each with `image_info` (optics, camera, scan_resolution), `channels` (per-channel `name`, `fluorophore`, `dye_from_name`, `exposure_time_us`, wavelengths, `color_rgb`, ...) and `scales` (per pyramid level).

`acquisition_format` One of `"brightfield"`, `"polaris_scanband"`, `"fusion_paged"`, `"ome_tiff"`, `"ome_zarr"`, or `"unknown"`.

Use the accessors (`qpi_format`, `qpi_channels`, `qpi_pixel_size_um`, `channel_table`, ...) rather than indexing the nested list directly.

Value

A `QPTIFFMetadata` object (a named list).

Examples

```
path <- system.file("extdata", "PA_HNC_sample.ome.tiff", package = "bgnormR")
img  <- read_qptiff(path)
meta <- metadata(img)
qpi_format(meta)
# Per-channel dye information as a tidy table
channel_table(meta)[, c("name", "fluorophore", "dye_from_name")]
```

names.QPTIFFImage — *Channel names of a QPTIFFImage (third-dimension names of the array)*

Description

Channel names of a QPTIFFImage (third-dimension names of the array)

Usage

```
## S3 method for class 'QPTIFFImage'
names(x)
```

Arguments

x	A QPTIFFImage.

Value

Character vector of channel names, or NULL.

names<-.QPTIFFImage — *Set the channel names of a QPTIFFImage*

Description

Replaces the third-dimension names (channel names) of a QPTIFFImage. Works for both eager (in-memory) and lazy (on-disk) objects.

Usage

```
## S3 replacement method for class 'QPTIFFImage'
names(x) <- value
```

Arguments

x	A QPTIFFImage.
value	Character vector of length equal to the number of channels.

Value

x with updated channel names.

Examples

```
arr <- array(1:8, dim = c(2, 2, 2),
             dimnames = list(NULL, NULL, c("ch1", "ch2")))
img <- as.QPTIFFImage(arr)
names(img) <- c("DAPI", "CD3")
names(img)
```

`OMEZarrArraySeed-class` *Seed class for lazy on-disk OME-Zarr access*

Description

Satisfies the **DelayedArray** seed contract (`dim`, `dimnames`, `type`, `extract_array`) for a single pyramid level of an OME-Zarr store, reading requested blocks through `Rarr::read_zarr_array`. Zarr axes (typically `c`, `y`, `x`) are permuted to the QPTIFFImage convention `[y, x, c]` = `[H, W, C]`. Do not construct directly; use `read_qptiff` on a `.ome.zarr` store.

Usage

```
## S4 method for signature 'OMEZarrArraySeed'
dim(x)

## S4 method for signature 'OMEZarrArraySeed'
dimnames(x)

## S4 method for signature 'OMEZarrArraySeed'
type(x)

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

## S4 method for signature 'OMEZarrArraySeed'
extract_array(x, index)
```

Arguments

`x`	An `OMEZarrArraySeed`.
`object`	An `OMEZarrArraySeed`.
`index`	List of length 3; each element is `NULL` (all indices) or an integer vector of 1-based indices for that dimension.

Value

`extract_array` returns an ordinary array; `dim` an integer vector; `dimnames` a list; `type` a character scalar.

Slots

`array_path` Absolute path to the level array inside the store.

`.dim` Integer `c(H, W, C)` in output order.

`.dimnames` List of dimnames; element 3 holds channel names.

`dtype` `"integer"` or `"double"`.

`axis_names` Character vector of zarr axis names (store order).

`zarr_channels` Integer; 1-based zarr channel index per output channel.

`divisor` Numeric; value pixels are divided by in `"double"` mode.

`metadata` A `QPTIFFMetadata` object.

`plot_distributions` *Per-marker intensity distribution with fitted GMM densities*

Description

Plots a histogram of log2-transformed pixel (or cell) intensities and overlays the component-wise density curves from the fitted GMM. One facet per marker. Accepts a `QPTIFFImage` returned by `bgnorm_pixels` (bgnorm results are carried as an attribute) or a `SummarizedExperiment` / `SpatialExperiment` with bgnorm results in `metadata(x)$bgnorm_results`. Multi-sample `SpatialExperiments` are automatically detected via `sample_id` in `colData` and produce facets by `sample_id` x `marker`.

Usage

```
plot_distributions(x, results = NULL, markers = NULL, ncol = NULL)
```

Arguments

`x`	A `QPTIFFImage` returned by `bgnorm_pixels` *or* a `SummarizedExperiment` / `SpatialExperiment`.
`results`	Optionally, the `QPTIFFImage` returned by `bgnorm_pixels` when `x` is the raw (unadjusted) `QPTIFFImage`. Ignored for `SummarizedExperiment` input.
`markers`	Character vector of markers to display, or `NULL` for all markers.
`ncol`	Number of facet columns.

Value

A `ggplot` object.

See Also

`plot_qptiff`, `plot_pixel_classes`, `bgnorm_pixels`, `bgnorm_sce`

Examples

```
path <- system.file("extdata", "PA_HNC_sample.ome.tiff", package = "bgnormR")
img  <- read_qptiff(path)
res  <- bgnorm_pixels(img, sample_prop = 0.1)
plot_distributions(res)
```

`plot_jsd_heatmap` *QC heatmap of Jensen-Shannon Divergence across markers and samples*

Description

Displays the per-marker JSD quality metric as a clustered heatmap, optionally annotated with the proportion of the signal GMM component. Accepts a named list of `BgnormResult` objects (single sample), a named list of such lists (multiple samples), or a `SummarizedExperiment` / `SpatialExperiment` with bgnorm results in `metadata(results)$bgnorm_results`.

Usage

```
plot_jsd_heatmap(
  results,
  cluster_rows = TRUE,
  cluster_cols = TRUE,
  show_tissue_positivity = TRUE
)
```

Arguments

`results`	A named list of `BgnormResult` objects (single sample), a named list of such lists (multiple samples), a named list of `QPTIFFImage` objects returned by `bgnorm_pixels` (one per sample; all must share the same channel names), a single `QPTIFFImage`, or a `SummarizedExperiment` / `SpatialExperiment`.
`cluster_rows`	Logical; cluster samples (rows)? Default `TRUE`.
`cluster_cols`	Logical; cluster markers (columns)? Default `TRUE`.
`show_tissue_positivity`	Logical; overlay tissue positivity as circles on the heatmap? Circle area is proportional to the tissue positivity ($\pi_3/(\pi_2+\pi_3)$ for three-component models; π_2 for two-component models). Circle colour indicates JSD quality: red (JSD < 0.1, low), orange (0.1-0.2, moderate), white ($\geq$ 0.2, good). Default `TRUE`.

Value

A `ggplot` object.

See Also

`bgnorm_pixels`, `bgnorm_sce`

Examples

```
path <- system.file("extdata", "PA_HNC_sample.ome.tiff", package = "bgnormR")
img  <- read_qptiff(path)
res  <- bgnorm_pixels(img, sample_prop = 0.1)
plot_jsd_heatmap(res)

# Multi-sample comparison (pass a named list)
plot_jsd_heatmap(list(sample_A = res, sample_B = res))
```

---

plot_pixel_classes *Spatial GMM class assignment plot for a QPTIFFImage or SpatialExperiment*

Description

Assigns each pixel/cell to its most probable GMM component and renders the result. For a `QPTIFFImage` produced by `bgnorm_pixels`: pseudocoloured raster (pixel-level data, three components). For a `SpatialExperiment`: scatter plot at spatial coordinates using the bgnorm results from `metadata(spe)$bgnorm_results` (two components). Multi-sample `SpatialExperiments` are automatically detected via the `sample_id` column in `colData` and faceted accordingly.

Usage

```
plot_pixel_classes(
  x,
  markers = NULL,
  resolution = 1L,
  point_size = 1,
  pixels = c(1024L, 1024L),
  flip_y = TRUE,
  large_data_threshold = 10000L,
  ncol = NULL
)
```

Arguments

`x`	A `QPTIFFImage` returned by `bgnorm_pixels` (contains both the adjusted pixel intensities and the embedded `BgnormResult` objects), *or* a `SpatialExperiment` with `metadata(x)$bgnorm_results` populated by `bgnorm_sce`.
`markers`	Character vector of markers to display, or `NULL` for all markers in `x`.
`resolution`	Positive integer; pixel downsample factor for raster output. Ignored for `SpatialExperiment`.
`point_size`	Base point size for `SpatialExperiment` scatter.
`pixels`	Integer `c(width, height)` for **scattermore** rasterisation. Default `c(1024L, 1024L)`.
`flip_y`	Logical; reverse y-axis for `SpatialExperiment`? Default `TRUE`.

`large_data_threshold`
: Cell count threshold for automatic **scattermore** use. Default `10000L`.

`ncol`
: Number of facet columns.

Value

A `ggplot` object.

See Also

`plot_qptiff`, `plot_distributions`, `bgnorm_pixels`, `bgnorm_sce`

Examples

```
path <- system.file("extdata", "PA_HNC_sample.ome.tiff", package = "bgnormR")
img  <- read_qptiff(path)
res  <- bgnorm_pixels(img, sample_prop = 0.1)
plot_pixel_classes(res, markers = c("PanCK", "CD20"))
```

---

plot_qptiff *Spatial intensity plot for a QPTIFFImage or SpatialExperiment*

Description

For a `QPTIFFImage`: renders up to 10 channels as an additive colour composite on a black background. Each channel is assigned a distinct colour; its per-pixel intensity (min-max normalised to [0, 1]) drives the channel's contribution - high-intensity pixels appear fully saturated while low-intensity pixels are transparent (black). Channels that overlap in space produce mixed additive colours (e.g. cyan + red -> white), matching the standard composite view in FIJI / napari. When no `markers` are provided the first 10 channels (by index) are displayed.

Usage

```
plot_qptiff(
  x,
  markers = NULL,
  resolution = 1L,
  palette = "magma",
  assay.type = "bgnorm",
  scale = c("marker", "sample", "none"),
  point_size = 1,
  pixels = c(1024L, 1024L),
  flip_y = TRUE,
  large_data_threshold = 10000L,
  ncol = NULL
)
```

Arguments

x	A QPTIFFImage *or* a SpatialExperiment.
markers	Character vector of channel names to display, or NULL to use the first 10 channels (for QPTIFFImage) or all features (for SpatialExperiment). At most 10 channels are shown for QPTIFFImage input; excess channels are dropped with a warning.
resolution	Positive integer; pixel downsample factor for QPTIFFImage input (ignored for SpatialExperiment).
palette	Viridis palette name used for SpatialExperiment input only. Default "magma".
assay.type	Assay to visualise when x is a SpatialExperiment. Default "bgnorm".
scale	Character; intensity scaling for QPTIFFImage input (ignored for SpatialExperiment). One of: "marker" (default) Per-channel min and 99.9th-percentile; each marker is stretched to full brightness independently. "sample" Global min and 99.9th-percentile computed across all channels in the image; preserves relative intensities between markers. "none" No scaling; values are clamped to [0, 1].
point_size	Base point size for SpatialExperiment scatter.
pixels	Integer c(width, height) for **scattermore** rasterisation. Default c(1024L, 1024L).
flip_y	Logical; reverse y-axis for SpatialExperiment? Default TRUE.
large_data_threshold	Cell count above which **scattermore** is used automatically. Default 10000L.
ncol	Ignored for QPTIFFImage (single composite panel). Number of facet columns for SpatialExperiment.

Details

For a SpatialExperiment: plots each cell at its spatial coordinates, coloured by its (per-channel scaled) intensity from the requested assay using a viridis colour scale and dark theme. When the cell count exceeds large_data_threshold and **scattermore** is installed, the scatter layer is rasterised automatically.

Value

A ggplot object.

See Also

plot_pixel_classes, plot_distributions, bgnorm_pixels, bgnorm_sce

Examples

```
path <- system.file("extdata", "PA_HNC_sample.ome.tiff", package = "bgnormR")
img  <- read_qptiff(path)
plot_qptiff(img, markers = c("PanCK", "CD20"))


# After normalisation: plot background-adjusted intensities
res <- bgnorm_pixels(img, sample_prop = 0.1)
plot_qptiff(res, markers = c("PanCK", "CD20"), scale = "sample")
```

`qc_summary` *Summarise QC across multiple bgnorm results*

Description

Summarise QC across multiple bgnorm results

Usage

```
qc_summary(results)
```

Arguments

`results`	A named list of `BgnormResult` objects (e.g., one per marker), or a `QPTIFFImage` returned by `bgnorm_pixels` (bgnorm results are extracted automatically).

Value

A `data.frame` with columns `name`, `jsd`, and `prop_signal` (proportion of pixels/cells in the signal component).

Examples

```
set.seed(2)
mk1 <- bgnorm_cells(c(exp(rnorm(300,log(300),0.5)), exp(rnorm(100,log(2000),0.9))))
mk2 <- bgnorm_cells(c(exp(rnorm(400,log(250),0.4)), exp(rnorm( 80,log(1800),0.8))))
qc_summary(list(CD3 = mk1, CD8 = mk2))
```

qpi_channel_names *Channel names of the primary scene*

Description

Channel names of the primary scene

Usage

```
qpi_channel_names(meta)
```

Arguments

meta	A QPTIFFMetadata object.

Value

Character vector of channel names (possibly empty).

Examples

```
path <- system.file("extdata", "PA_HNC_sample.ome.tiff", package = "bgnormR")
qpi_channel_names(metadata(read_qptiff(path)))
```

qpi_channels *Per-channel metadata of the primary scene*

Description

Per-channel metadata of the primary scene

Usage

```
qpi_channels(meta)
```

Arguments

meta	A QPTIFFMetadata object.

Value

A list of per-channel metadata lists (possibly empty).

Examples

```
path <- system.file("extdata", "PA_HNC_sample.ome.tiff", package = "bgnormR")
length(qpi_channels(metadata(read_qptiff(path))))
```

qpi_format *Acquisition format of a parsed image*

Description

Acquisition format of a parsed image

Usage

```
qpi_format(meta)
```

Arguments

meta	A QPTIFFMetadata object (or legacy metadata list).

Value

Character scalar, or NULL.

Examples

```
path <- system.file("extdata", "PA_HNC_sample.ome.tiff", package = "bgnormR")
meta <- metadata(read_qptiff(path))
qpi_format(meta)
```

qpi_is_brightfield *Is the primary scene a brightfield (RGB) image?*

Description

Is the primary scene a brightfield (RGB) image?

Usage

```
qpi_is_brightfield(meta)
```

Arguments

meta	A QPTIFFMetadata object.

Value

Logical scalar.

Examples

```
path <- system.file("extdata", "PA_HNC_sample.ome.tiff", package = "bgnormR")
qpi_is_brightfield(metadata(read_qptiff(path)))
```

qpi_n_levels *Number of pyramid levels of the primary scene*

Description

Number of pyramid levels of the primary scene

Usage

```
qpi_n_levels(meta)
```

Arguments

meta	A QPTIFFMetadata object.

Value

Integer scalar (at least 1).

Examples

```
path <- system.file("extdata", "PA_HNC_sample.ome.tiff", package = "bgnormR")
qpi_n_levels(metadata(read_qptiff(path)))
```

qpi_pixel_size_um *Physical pixel size (microns) of the primary scene, full resolution*

Description

Physical pixel size (microns) of the primary scene, full resolution

Usage

```
qpi_pixel_size_um(meta)
```

Arguments

meta	A QPTIFFMetadata object.

Value

Numeric scalar, or NULL.

Examples

```
path <- system.file("extdata", "PA_HNC_sample.ome.tiff", package = "bgnormR")
qpi_pixel_size_um(metadata(read_qptiff(path)))
```

qpi_primary_scene *Primary scene of a parsed image (FullResolution if present, else the first)*

Description

Primary scene of a parsed image (FullResolution if present, else the first)

Usage

```
qpi_primary_scene(meta)
```

Arguments

meta	A QPTIFFMetadata object.

Value

A scene list, or NULL when no scenes are present.

Examples

```
path <- system.file("extdata", "PA_HNC_sample.ome.tiff", package = "bgnormR")
scene <- qpi_primary_scene(metadata(read_qptiff(path)))
scene$image_info$image_type
```

QPTIFFArraySeed-class *Seed class for lazy on-disk QPTIFF access*

Description

An S4 class that satisfies the **DelayedArray** seed contract, enabling lazy, block-based processing of large Akoya PhenoCycler-Fusion QPTIFF images. Each channel (TIFF page) is read from disk only when data for that channel is actually requested by the **DelayedArray** framework.

Called internally by the **DelayedArray** framework. Reads only the requested channels and the spatial region specified by index from disk, keeping memory use proportional to the data actually needed.

Usage

```
## S4 method for signature 'QPTIFFArraySeed'
dim(x)

## S4 method for signature 'QPTIFFArraySeed'
dimnames(x)

## S4 method for signature 'QPTIFFArraySeed'
type(x)

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

## S4 method for signature 'QPTIFFArraySeed'
extract_array(x, index)
```

Arguments

x	A QPTIFFArraySeed.
object	A QPTIFFArraySeed.
index	List of length 3. Each element is either NULL (all indices for that dimension) or a sorted integer vector of 1-based indices.

Details

Internally the seed reads the full TIFF/BigTIFF directory chain at construction time (fast: only header bytes, no pixel data) and stores tile/strip layout metadata for each channel page. Pixel data is fetched per-tile (when the image is tiled) or per-strip with DEFLATE/zlib decompression handled by R's built-in memDecompress. Unsupported compression types fall back to tiff::readTIFF.

Do not construct this class directly; use read_qptiff(path, lazy = TRUE).

Value

An ordinary array with dimensions c(length(i_rows), length(i_cols), length(i_channels)).

Slots

filepath Absolute path to the QPTIFF file.

.dim Integer vector [H, W, C].

.dimnames List of dimnames; element 3 holds channel names.

page_layouts List of per-channel page layout objects (one per channel) each containing the TIFF structural metadata needed to fetch that page.

dtype Character, "integer" (raw 16-bit values 0-65535) or "double" (normalised to [0, 1]).

metadata List; rich QPI metadata as returned by .parse_qpi_xml.

level Integer; pyramid resolution level (1 = full resolution).

QPTIFFImage — *QPTIFFImage: an in-memory or on-disk multi-channel image*

Description

An S3 class representing a `[H x W x C]` multiplex image, either eagerly loaded as a plain 3-D array or lazily backed by a `DelayedArray` (see `QPTIFFArraySeed-class`). Created by `read_qptiff` or `as.QPTIFFImage`, and consumed by `bgnorm_pixels`, `plot_qptiff`, and `write_qptiff`.

Value

Not applicable; documents the `QPTIFFImage` class structure.

QPTIFFMetadata — *QPTIFFMetadata: an OME-organised image metadata object*

Description

An S3-classed nested list returned by `metadata` on a `QPTIFFImage`. It mirrors the OME hierarchy: a slide-constant `slide` block, a list of image `images` (each with `image_info`, `channels`, and `scales`), an `acquisition_format`, and the raw source `raw_xml`. Prefer the accessor helpers - `qpi_format`, `qpi_channels`, `qpi_channel_names`, `qpi_pixel_size_um`, `qpi_n_levels`, `qpi_primary_scene`, `qpi_is_brightfield`, `channel_table` - over indexing the nested list directly.

Value

Not applicable; documents the `QPTIFFMetadata` structure.

See Also

`metadata`, `read_qptiff`

read_qptiff — *Read a multiplex image (QPTIFF, OME-TIFF, or OME-Zarr)*

Description

Reads a multiplex image into R without Java, auto-detecting the container:

- **QPTIFF** from Akoya PhenoCycler-Fusion (formerly CODEX), Cell DIVE, or Vectra / Polaris scanners - brightfield RGB, Polaris ScanBand XML, and Fusion per-page JSON+XML variants.
- **OME-TIFF** (as written by tifffile, Bio-Formats, or QuPath) - channel names are read from the `<Channel Name="...">` attributes of the OME-XML.
- **OME-Zarr** (OME-NGFF) stores - requires the **Rarr** package; channel names / colours are read from `omero.channels`.

Usage

```
read_qptiff(path, channels = NULL, level = 1L, as_integer = TRUE, lazy = FALSE)
```

Arguments

path	Character, path to the image. A QPTIFF / OME-TIFF file, or an OME-Zarr store (a directory, or a path ending in .zarr).
channels	Character vector of channel names to load, an integer vector of 1-based channel indices, or NULL (default) to load all channels.
level	Integer, pyramid resolution level. 1 = full resolution (default), 2 = half resolution, etc. OME-TIFF sub-resolution pyramids (SubIFDs) are not supported; use level = 1.
as_integer	Logical; return raw 16-bit integers (0-65535) rather than normalised [0, 1] doubles? Default TRUE.
lazy	Logical; if TRUE return a DelayedArray backed by a QPTIFFArraySeed that reads pages from disk on demand. If FALSE (default) load all requested channels into memory and return a QPTIFFImage.

Details

Channel names and rich per-channel metadata (fluorophore, exposure time, wavelengths, filters, colour) are parsed into an OME-organised QPTIFFMetadata object, available via metadata.

The native, Java-free TIFF/QPTIFF reader and writer implemented here was translated from the bioio-tifffile fork by Rafael Tubelleza.

Value

lazy = FALSE A QPTIFFImage - a 3-D numeric array [height, width, channels] with class c("QPTIFFImage", "array"). Channel names are stored in dimnames(img)[[3]]. Standard array subscripting works: img[, , "DAPI"] extracts a single channel as a matrix; img[1:512, 1:512,] crops spatially. Rich metadata is in attr(img, "metadata").

lazy = TRUE A DelayedArray with dim = c(H, W, C). Individual channel pages are read from disk only when accessed. The seed is a QPTIFFArraySeed accessible via DelayedArray::seed(arr).

Examples

```
path <- system.file("extdata", "PA_HNC_sample.ome.tiff", package = "bgnormR")
img  <- read_qptiff(path)
dim(img)                      # c(H, W, n_channels)
names(img)                    # channel names
cd20 <- img[, , "CD20"]      # extract one channel as a 2-D matrix

# Load specific channels only
img2 <- read_qptiff(path, channels = c("CD20", "PanCK"))
names(img2)


# Lazy / out-of-core load (backed by DelayedArray)
```

```
arr <- read_qptiff(path, lazy = TRUE)
```

`write_qptiff` *Write a QPTIFFImage to a TIFF file*

Description

Writes a `QPTIFFImage` as a multi-page 16-bit grayscale OME-TIFF (one page per channel). The first page carries an OME-XML `ImageDescription` (tag 270) that declares the pages as *channels* (`SizeC` = `nChannels`, `SizeZ` = `SizeT` = 1), so viewers such as QuPath / Bio-Formats interpret each page as a channel rather than a timepoint. Per-channel bgnorm results are embedded so the file is self-documenting and round-trips through `read_qptiff`.

Usage

```
write_qptiff(x, path)
```

Arguments

`x`	A `QPTIFFImage` (eager or lazy).
`path`	Output file path (character scalar). The directory must exist.

Value

`path`, invisibly.

Intensity transform

bgnorm-adjusted images The QPTIFFImage stores background-adjusted $\log_2$-intensities. These are inverted with 2^x before writing so the output values are in a linear intensity scale.

Raw images (no bgnorm results) Written as-is; values are assumed to be in `[0, 65535]`.

All output values are rounded to the nearest integer and clamped to the `[0, 65535]` range before being written as 16-bit unsigned integers.

The native, Java-free TIFF/QPTIFF reader and writer implemented here was translated from the bioio-tifffile fork by Rafael Tubelleza.

Metadata

The first page's `ImageDescription` tag (TIFF tag 270) is an OME-XML document. Its `<Pixels>` element declares `SizeC` = `nChannels` with one `<Channel Name="...">` per channel and an explicit `<TiffData>` page-to-channel mapping, which is what makes downstream viewers read the pages as channels. Channel 0's bgnorm metadata is appended to the OME root as no-namespace elements (ignored by OME readers but read back by `read_qptiff`); pages 1..n each carry a minimal `PerkinElmerQPI` block:

```
<PerkinElmerQPI>
  <Biomarker>CD20</Biomarker>
  <transform>2^x</transform>
  <bgnorm>{"level":"pixel","jsd":0.35,...}</bgnorm>
</PerkinElmerQPI>
```

The <Biomarker> element is the channel name; <bgnorm> holds a JSON object with GMM parameters, JSD, threshold, and normalisation flags (only present for bgnorm-adjusted images). If **jsonlite** is not installed, the <bgnorm> element is omitted.

See Also

read_qptiff, bgnorm_pixels

Examples

```
path <- system.file("extdata", "PA_HNC_sample.ome.tiff", package = "bgnormR")
img  <- read_qptiff(path)
res  <- bgnorm_pixels(img, sample_prop = 0.1)
out  <- file.path(tempdir(), "PA_HNC_bgnorm.tif")
write_qptiff(res, out)

# Round-trip: channel names and dimensions are preserved
img2 <- read_qptiff(out)
names(img2)
dim(img2)
```

Index

[.QPTIFFImage, 4

as.array.QPTIFFImage, 5
as.list.QPTIFFImage, 5
as.matrix, *6*
as.matrix.QPTIFFImage, 6
as.QPTIFFImage, 6, *9, 10*, *32*

bgnorm_cells, *3*, 7, *10–12*, *15*
bgnorm_pixels, *3*, 8, *8*, *10–12*, *15*, *21–26*, *32*, *35*
bgnorm_results, *9*, 10, *10*, *12*
bgnorm_sce, *3*, *8*, 11, *21–25*
bgnormR *(*bgnormR-package*)*, 3
bgnormR-package, 3
BgnormResult, *9, 10*, 12, *15*, *23*
BiocParallelParam, *9*, *11*

channel_table, 13, *18*, *32*

DelayedArray, *33*
dim,OMEZarrArraySeed-method *(*OMEZarrArraySeed-class*)*, 20
dim,QPTIFFArraySeed-method *(*QPTIFFArraySeed-class*)*, 30
dimnames,OMEZarrArraySeed-method *(*OMEZarrArraySeed-class*)*, 20
dimnames,QPTIFFArraySeed-method *(*QPTIFFArraySeed-class*)*, 30
dimnames<-.QPTIFFImage, 13

extract_array,OMEZarrArraySeed-method *(*OMEZarrArraySeed-class*)*, 20
extract_array,QPTIFFArraySeed-method *(*QPTIFFArraySeed-class*)*, 30

inv_log_transform, 14, *17*

jsd_gaussians, 15
jsd_qc, *3*, *10*, 15

length.QPTIFFImage, 16
log_transform, *14*, 16

median_filter_3x3, 17
metadata, 18, *18*, *32, 33*

names.QPTIFFImage, 19
names<-.QPTIFFImage, 19

OMEZarrArraySeed-class, 20

plot_distributions, *3*, *12*, 21, *24, 25*
plot_jsd_heatmap, *3*, 22
plot_pixel_classes, *3*, *21*, 23, *25*
plot_qptiff, *3*, *21*, 24, *24*, *32*

qc_summary, 26
qpi_channel_names, 27, *32*
qpi_channels, *18*, 27, *32*
qpi_format, *18*, 28, *32*
qpi_is_brightfield, 28, *32*
qpi_n_levels, 29, *32*
qpi_pixel_size_um, *18*, 29, *32*
qpi_primary_scene, 30, *32*
QPTIFFArraySeed, *33*
QPTIFFArraySeed-class, 30
QPTIFFImage, *8–10*, *18*, *21–25*, 32, *32–34*
QPTIFFMetadata, *18*, 32, *33*

read_qptiff, *3*, *20*, *31*, 32, *32*, *34, 35*

show,OMEZarrArraySeed-method *(*OMEZarrArraySeed-class*)*, 20
show,QPTIFFArraySeed-method *(*QPTIFFArraySeed-class*)*, 30
SpatialExperiment, *21–23*, *25*
SummarizedExperiment, *21, 22*

type,OMEZarrArraySeed-method *(*OMEZarrArraySeed-class*)*, 20

type,QPTIFFArraySeed-method
(QPTIFFArraySeed-class), 30

write_qptiff, *32*, 34