Package: hyperspectaculR (via r-universe)

November 3, 2026

Title Publication-Grade Artistic Visualization of Hyperspectral Imagery

Version 0.2.0.9001

Description Creates artistic compositions and descriptive summaries from hyperspectral imaging cubes or numeric arrays. Provides radial spectral mandalas, cumulative spectral change, multi-band fusions, exact spectral density histograms, and experimental RMS spectral slope and within-pixel spectral quartiles. Returns composable 'ggplot' objects with explicit validity rules, display domains and compact original-rendering provenance. Does not read recordings, calibrate input values or infer physiology.

License MIT + file LICENSE

URL https://github.com/CTTIR/hyperspectaculR, https://cttir.github.io/hyperspectaculR/

BugReports https://github.com/CTTIR/hyperspectaculR/issues

Encoding UTF-8

Language en-US

Roxygen list(markdown = TRUE)

Depends R (>= 4.1.0)

Imports cli, ggplot2 (>= 3.4.0), grDevices, rlang, scales, stats, viridisLite (>= 0.4.0)

Suggests hyperspectR, tivis.r, patchwork, testthat (>= 3.0.0), vdiffr, withr, knitr, rmarkdown, covr

Additional_repositories https://cttir.r-universe.dev

Config/testthat/edition 3

VignetteBuilder knitr

Config/Needs/website CTTIR/themakR

Config/roxygen2/version 8.0.0

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

Date/Publication 2026-11-03 10:59:14 UTC

RemoteUrl https://github.com/CTTIR/hyperspectaculR

RemoteRef HEAD

RemoteSha d8fa8158b8cdf9f8f52fd13b2b515d44da4b0d74

Contents

hsa_demo_cube 2
hsa_fusion 3
hsa_mandala 4
hsa_palette 5
hsa_provenance 6
hsa_spectral_density 7
hsa_spectral_flux 8
hsa_spectral_gradient 10
hsa_spectral_quartiles 11
hsa_theme 12

Index **14**

`hsa_demo_cube` *A Small Demonstration Cube*

Description

A deterministic synthetic cube for examples and tests, shaped like an `hsi_cube` so the art functions can be demonstrated without **hyperspectR** or any recorded data.

Usage

```
hsa_demo_cube(rows = 48L, cols = 64L, bands = 24L, seed = 42L)
```

Arguments

`rows`, `cols`	Spatial dimensions. Default `48` by `64`.
`bands`	Number of bands. Default `24`.
`seed`	Random seed. Default `42`.

Value

A list with `data` and `wavelengths`, classed as `hsi_cube`.

Examples

```
cube <- hsa_demo_cube()
dim(cube$data)
```

hsa_fusion *Multi-Band Spectral Fusion*

Description

Builds a colour composite by equally averaging three groups of measured bands. Averaging may reduce independent band noise but does not calibrate values. Explicit groups may overlap; singleton groups are allowed, including identical channels from one band. Each channel uses its own stretch.

Usage

```
hsa_fusion(
  cube,
  red = NULL,
  green = NULL,
  blue = NULL,
  by = c("index", "wavelength"),
  stretch = c("percentile", "range", "none"),
  probs = c(0.02, 0.98),
  display_limits = c(0, 1),
  value_label = "input value",
  interpolate = TRUE,
  missing = c("propagate", "available")
)
```

Arguments

cube	An hsi_cube (from **hyperspectR**) or a 3-D array with dimensions (rows, cols, bands), of integer or double values. Optional wavelengths metadata must be a numeric vector, finite, unique and strictly increasing in nm. Without it, coordinates are band indices. An optional logical spatial mask has dimensions (rows, cols) and no missing entries; FALSE excludes a pixel. Arrays may carry these as attributes; an hsi_cube may carry them as exactly named list fields. Dimension names do not alter spatial sizes. Nonfinite observations are excluded.
red, green, blue	Integer vectors of band indices, or numeric wavelengths in nm when by = "wavelength". NULL (default) splits the spectrum into three contiguous thirds, long wavelengths to red.
by	Either "index" (default) or "wavelength".
stretch	Contrast stretch: "percentile" (default), "range" or "none".
probs	Percentiles for stretch = "percentile". Default c(0.02, 0.98).
display_limits	Two increasing finite endpoints used as the fixed colour domain for stretch = "none". Default c(0, 1).
value_label	A truthful label for the input values. Default "input value". A reflectance label requires known upstream calibration; numeric range alone cannot establish calibration.

`interpolate`	Logical; interpolate raster pixels. Default `TRUE`.
`missing`	Missing-band policy. `"propagate"` (default) requires every selected band; `"available"` averages the finite contributors in a group.

Value

A **ggplot2** object.

Examples

```
cube <- hsa_demo_cube()
hsa_fusion(cube)
```

`hsa_mandala` *Radial Spectral Mandala*

Description

Renders the scene as concentric annuli, each drawn from a sampled spectral band: the innermost ring uses the first band and the outermost uses the last. The result is a spatial composition; unequal annular areas do not estimate the spectral distribution. A single ring selects the first band.

Usage

```
hsa_mandala(
  cube,
  centre = NULL,
  n_rings = 36L,
  palette = "magma",
  stretch = c("percentile", "range", "none"),
  probs = c(0.02, 0.98),
  display_limits = c(0, 1),
  value_label = "input value",
  interpolate = TRUE,
  sampling = c("index", "wavelength")
)
```

Arguments

`cube`	An `hsi_cube` (from **hyperspectR**) or a 3-D array with dimensions `(rows, cols, bands)`, of integer or double values. Optional `wavelengths` metadata must be a numeric vector, finite, unique and strictly increasing in nm. Without it, coordinates are band indices. An optional logical spatial `mask` has dimensions `(rows, cols)` and no missing entries; `FALSE` excludes a pixel. Arrays may carry these as attributes; an `hsi_cube` may carry them as exactly named list fields. Dimension names do not alter spatial sizes. Nonfinite observations are excluded.

centre	Numeric length-2 vector `c(x, y)` in pixels. `NULL` (default) uses the image centre.
n_rings	Number of annuli. Default `36`. More rings sample the spectrum more finely; fewer give bolder banding.
palette	Palette name passed to `hsa_palette()`.
stretch	Contrast stretch: `"percentile"` (default), `"range"` or `"none"`.
probs	Percentiles for `stretch = "percentile"`. Default `c(0.02, 0.98)`.
display_limits	Two increasing finite endpoints used as the fixed colour domain for `stretch = "none"`. Default `c(0, 1)`.
value_label	A truthful label for the input values. Default `"input value"`. A reflectance label requires known upstream calibration; numeric range alone cannot establish calibration.
interpolate	Logical; interpolate raster pixels. Default `TRUE`.
sampling	Ring targets: `"index"` (default) or `"wavelength"`. Wavelength sampling requires physical metadata in nm.

Details

This uses the spectral dimension for a spatial composition. A single band or repeated ring selections can produce a degenerate single-band result. The default radius-to-band-index mapping is linear. Wavelength sampling uses equally spaced physical targets and nearest measured bands, with ties toward shorter wavelengths. Repeated selections and empty rings retain the geometry and are disclosed in the caption and provenance.

Value

A **ggplot2** object.

Examples

```
cube <- hsa_demo_cube()
hsa_mandala(cube, n_rings = 12)
```

---

hsa_palette *Perceptually Uniform Palettes for Spectral Imagery*

Description

Returns colour ramps suitable for scientific imagery. The palettes other than `"turbo"` are perceptually uniform and monotonic in lightness, so ordering survives greyscale printing and remains legible to colourblind readers. `"turbo"` is retained with a warning for compatibility.

Usage

```
hsa_palette(
 name = c("viridis", "magma", "inferno", "plasma", "cividis", "mako", "rocket", "turbo"),
  n = 256L,
  direction = 1
)
```

Arguments

name	Palette name. One of "viridis", "magma", "inferno", "plasma", "cividis", "mako", "rocket", "turbo".
n	Number of colours. Default 256.
direction	1 (default) or -1 to reverse.

Value

A character vector of hex colours.

Examples

```
head(hsa_palette("magma", n = 5))
```

---

hsa_provenance *Inspect Original Rendering Provenance*

Description

Returns the compact calculation and display record attached when a hyperspectaculR plot was created. Later arbitrary ggplot modifications may change the visible plot without updating this original-rendering record. Ordinary + ggplot2::theme() and + ggplot2::labs() additions preserve it. Inspect individual component plots before assembling a patchwork; a combined patchwork has no single calculation record. Full actual band selections, endpoints, fallback reasons, exclusions and quantity definitions remain inspectable even when plot labels abbreviate them. Source cubes and arbitrary input metadata are not retained in the record.

Usage

```
hsa_provenance(plot)
```

Arguments

plot	A plot returned by a hyperspectaculR renderer.

Value

A compact named list describing the original rendering.

`hsa_spectral_density` *Spectral Density*

Description

Draws the joint distribution of input values and spectral coordinates as an exact two-dimensional histogram. A binned mean and binned 5–95% pixel envelope are calculated from the retained histogram counts.

Usage

```
hsa_spectral_density(
  cube,
  nbins = 128L,
  limits = NULL,
  transform = c("log1p", "identity"),
  normalise = c("none", "band"),
  palette = "mako",
  show_limits = FALSE,
  probs = c(0.02, 0.98),
  value_label = "input value"
)
```

Arguments

`cube`	An `hsi_cube` (from **hyperspectR**) or a 3-D array with dimensions `(rows, cols, bands)`, of integer or double values. Optional `wavelengths` metadata must be a numeric vector, finite, unique and strictly increasing in nm. Without it, coordinates are band indices. An optional logical spatial `mask` has dimensions `(rows, cols)` and no missing entries; `FALSE` excludes a pixel. Arrays may carry these as attributes; an `hsi_cube` may carry them as exactly named list fields. Dimension names do not alter spatial sizes. Nonfinite observations are excluded.
`nbins`	Number of value bins. Must be an integer of at least two. Default `128`.
`limits`	Numeric length-two value range to bin over. `NULL` (default) uses the 0.1st and 99.9th percentiles of all finite eligible values.
`transform`	Display transform: `"log1p"` (default) or `"identity"`. Legend labels are returned to raw counts or shares.
`normalise`	`"none"` (default) displays raw counts; `"band"` displays shares of the retained count in each band. Empty bands remain missing.
`palette`	Palette name passed to `hsa_palette()`.
`show_limits`	Logical. Draw global raw eligible-value percentiles using `probs`. These references include finite observations outside `limits` and are separate from image enhancement limits. Default `FALSE`.
`probs`	Two increasing finite probabilities between zero and one for the global raw eligible-value references when `show_limits = TRUE`. These do not set the histogram range or image enhancement limits.

`value_label` A truthful label for the input values. Default `"input value"`. A reflectance label requires known upstream calibration; numeric range alone cannot establish calibration.

Details

Values outside `limits` are dropped rather than clipped into the end bins. Masks and nonfinite values are excluded. The overlay describes the retained pixel distribution; it is not a confidence interval.

Value

A **ggplot2** object with compact original-rendering provenance.

Examples

```
cube <- hsa_demo_cube()
hsa_spectral_density(cube)
```

---

hsa_spectral_flux *Cumulative Spectral Change*

Description

Sums the absolute change between consecutive bands at every pixel, giving a field of total spectral variability. Flat spectra appear dark; pixels whose input values change across the spectrum appear bright.

Usage

```
hsa_spectral_flux(
  cube,
  palette = "inferno",
  normalise = TRUE,
  stretch = c("percentile", "range", "none"),
  probs = c(0.02, 0.98),
  display_limits = c(0, 1),
  value_label = "input value",
  interpolate = TRUE,
  normalization = NULL
)
```

Arguments

`cube` An `hsi_cube` (from **hyperspectR**) or a 3-D array with dimensions `(rows, cols, bands)`, of integer or double values. Optional `wavelengths` metadata must be a numeric vector, finite, unique and strictly increasing in nm. Without it, coordinates are band indices. An optional logical spatial `mask` has dimensions `(rows, cols)`

and no missing entries; FALSE excludes a pixel. Arrays may carry these as attributes; an hsi_cube may carry them as exactly named list fields. Dimension names do not alter spatial sizes. Nonfinite observations are excluded.

palette — Palette name passed to hsa_palette().

normalise — Logical. Divide by the number of band steps, so the value is a mean absolute step rather than a total. Default TRUE.

stretch — Contrast stretch: "percentile" (default), "range" or "none".

probs — Percentiles for stretch = "percentile". Default c(0.02, 0.98).

display_limits — Two increasing finite endpoints used as the fixed colour domain for stretch = "none". Default c(0, 1).

value_label — A truthful label for the input values. Default "input value". A reflectance label requires known upstream calibration; numeric range alone cannot establish calibration.

interpolate — Logical; interpolate raster pixels. Default TRUE.

normalization — NULL preserves normalise; otherwise "total", "mean_step", or "wavelength_span" selects V, V/(B-1), or V/L, where V = sum(abs(diff(x))) and L is the endpoint wavelength span. Contradictory explicit normalise and normalization settings are rejected.

Details

Unlike a single-band image this cannot be produced from any one wavelength, and unlike a variance map it is sensitive to the ordering of the bands, so it responds to spectral shape rather than spread alone.

Wavelength-span normalization requires at least two physical wavelengths in nm and a finite positive representable span. It measures mean absolute slope of the piecewise-linear spectrum in input-value units per nm. Gaps bridge unobserved structure; this does not correct sampling density, unresolved peaks, or measurement noise. Every band must be finite at a valid pixel. Nonrepresentable requested arithmetic raises an error.

Value

A **ggplot2** object.

Examples

```
cube <- hsa_demo_cube()
hsa_spectral_flux(cube)
```

`hsa_spectral_gradient` *RMS Spectral Slope*

Description

An experimental descriptive image of the nonnegative RMS magnitude of the wavelength derivative of each pixel's piecewise-linear spectrum.

Usage

```
hsa_spectral_gradient(
  cube,
  palette = "inferno",
  stretch = c("percentile", "range", "none"),
  probs = c(0.02, 0.98),
  display_limits = c(0, 1),
  value_label = "input value",
  interpolate = TRUE
)
```

Arguments

`cube`	An `hsi_cube` (from **hyperspectR**) or a 3-D array with dimensions `(rows, cols, bands)`, of integer or double values. Optional `wavelengths` metadata must be a numeric vector, finite, unique and strictly increasing in nm. Without it, coordinates are band indices. An optional logical spatial `mask` has dimensions `(rows, cols)` and no missing entries; `FALSE` excludes a pixel. Arrays may carry these as attributes; an `hsi_cube` may carry them as exactly named list fields. Dimension names do not alter spatial sizes. Nonfinite observations are excluded.
`palette`	Palette name passed to `hsa_palette()`.
`stretch`	Contrast stretch: `"percentile"` (default), `"range"` or `"none"`.
`probs`	Percentiles for `stretch = "percentile"`. Default `c(0.02, 0.98)`.
`display_limits`	Two increasing finite endpoints used as the fixed colour domain for `stretch = "none"`. Default `c(0, 1)`.
`value_label`	A truthful label for the input values. Default `"input value"`. A reflectance label requires known upstream calibration; numeric range alone cannot establish calibration.
`interpolate`	Logical; interpolate raster pixels. Default `TRUE`.

Details

For interval widths h = diff(wavelengths) and endpoint span L, G = sqrt(sum(h * (diff(x)/h)^2)/L). Values have input-value units per nm. This is neither a signed derivative, a spatial gradient, nor uncertainty. Short intervals and measurement noise can dominate the result. Gaps in sampling bridge unresolved structure; no smoothing or noise correction is applied. Physical wavelength metadata

in nm and at least two bands are required, with finite positive representable intervals and span. A masked or incomplete spectrum is transparent; missing bands are never bridged. Nonrepresentable requested arithmetic raises an error.

Value

A ggplot object with raw values and original-rendering provenance.

Examples

```
hsa_spectral_gradient(hsa_demo_cube())
```

`hsa_spectral_quartiles` *Within-Pixel Spectral Quartiles*

Description

An experimental descriptive composition with equal-band type-7 quartiles computed over the measured bands within each spatial pixel.

Usage

```
hsa_spectral_quartiles(
  cube,
  palette = "magma",
  stretch = c("percentile", "range", "none"),
  probs = c(0.02, 0.98),
  display_limits = c(0, 1),
  value_label = "input value",
  interpolate = TRUE
)
```

Arguments

`cube`	An `hsi_cube` (from **hyperspectR**) or a 3-D array with dimensions `(rows, cols, bands)`, of integer or double values. Optional `wavelengths` metadata must be a numeric vector, finite, unique and strictly increasing in nm. Without it, coordinates are band indices. An optional logical spatial `mask` has dimensions `(rows, cols)` and no missing entries; `FALSE` excludes a pixel. Arrays may carry these as attributes; an `hsi_cube` may carry them as exactly named list fields. Dimension names do not alter spatial sizes. Nonfinite observations are excluded.
`palette`	Palette name passed to `hsa_palette()`.
`stretch`	Contrast stretch: `"percentile"` (default), `"range"` or `"none"`.
`probs`	Percentiles for `stretch = "percentile"`. Default `c(0.02, 0.98)`.
`display_limits`	Two increasing finite endpoints used as the fixed colour domain for `stretch = "none"`. Default `c(0, 1)`.

value_label	A truthful label for the input values. Default "input value". A reflectance label requires known upstream calibration; numeric range alone cannot establish calibration.
interpolate	Logical; interpolate raster pixels. Default TRUE.

Details

The panels are the 25th percentile, median and 75th percentile. Every band must be finite and the pixel unmasked; validity is shared by all panels. One band is allowed and gives three identical values. Bands have equal weight, regardless of wavelength spacing. These summaries omit spectral ordering, depend on sampling density, and are not confidence intervals or uncertainty estimates. A single stretch pools the three rendered raw panel values; all panels share its domain and legend. With a stretch the legend labels display values on 0-1; with stretch = "none" it labels raw input values in the supplied display_limits domain.

Value

One faceted ggplot object, retaining raw panel values and provenance.

Examples

```
hsa_spectral_quartiles(hsa_demo_cube())
```

---

hsa_theme *A Theme for Hyperspectral Figures*

Description

A dark, chrome-free theme for image-like panels. Axes, grids and panel backgrounds are removed, because for a rendered scene they carry no information and compete with the image.

Usage

```
hsa_theme(
  base_size = 11,
  base_family = "",
  background = "#0B0B0F",
  ink = "#E8E8EC"
)
```

Arguments

base_size	Base font size in points. Default 11.
base_family	Font family. Default "" (the device default), which keeps figures portable; set it explicitly when embedding fonts for a journal.
background	Panel and plot background colour. Default "#0B0B0F".
ink	Foreground colour for text. Default "#E8E8EC".

Details

Deliberately standalone: it does not require **hyperspectR**, so figures can be produced anywhere a cube is available, while remaining visually compatible with `hyperspectR::theme_hsi()`.

Value

A **ggplot2** theme object.

Examples

```
library(ggplot2)
ggplot(data.frame(x = 1, y = 1), aes(x, y)) + geom_point() + hsa_theme()
```

Index

hsa_demo_cube, 2
hsa_fusion, 3
hsa_mandala, 4
hsa_palette, 5
hsa_palette(), *5*, *7*, *9–11*
hsa_provenance, 6
hsa_spectral_density, 7
hsa_spectral_flux, 8
hsa_spectral_gradient, 10
hsa_spectral_quartiles, 11
hsa_theme, 12