Package: bloodspottR (via r-universe)

November 3, 2026

Title Calibrated Histology Burden and Erythrocyte Profile Analysis

Version 0.1.0.9000

Description Tools for physically calibrated histology burden summaries, explicit measurement-area accounting, and tissue-aware spatial analysis. Supports reproducible development of red-chromogen and erythrocyte-profile workflows. Image-derived candidates require independent validation before biological interpretation.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.2.0)

Imports cli, digest, jsonlite

Suggests testthat (>= 3.0.0), sf, spatstat.geom, shiny, processx, ps, openxlsx2, cellspecR, knitr, rmarkdown, covr

Remotes CTTIR/cellspecR

VignetteBuilder knitr

Roxygen list(markdown = TRUE)

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

URL https://github.com/CTTIR/bloodspottR

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

Config/Needs/website CTTIR/themakR, pkgdown

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

Date/Publication 2026-11-03 11:17:50 UTC

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

RemoteRef HEAD

RemoteSha a8cfcd205233b980aa8b59768fcc704b71137d38

Contents

aggregate_burden . . . 2
aggregate_event_burden . . . 3
bs_analyze . . . 4
bs_app . . . 4
bs_as_cellspec . . . 5
bs_backend . . . 6
bs_color_features . . . 7
bs_compare . . . 8
bs_cttir_inventory . . . 8
bs_example . . . 9
bs_export_results . . . 9
bs_import_legacy . . . 10
bs_profile_extent . . . 10
bs_project . . . 11
bs_report . . . 12
bs_review_confirm . . . 12
bs_review_plan . . . 13
bs_run_backend . . . 14
bs_status . . . 14
bs_summarize . . . 15
bs_train . . . 15
bs_validate . . . 16
bs_viewer_example . . . 17
cell_pattern . . . 17
measure_burden . . . 18
measure_event_burden . . . 19
tissue_window . . . 20

Index **21**

`aggregate_burden` *Aggregate disjoint measurements using summed areas and counts*

Description

Each `physical_area_id` must occur once. IDs are a safeguard, not proof that two geometries are disjoint; scan overlap must be reconciled upstream. Missing profile counts propagate to the aggregate rather than becoming zero. The default produces descriptive slide-level summaries without requiring animal identities or treatment labels. Unselected metadata are not returned. A slide summary does not establish an independent biological replicate. Optional `qc_flags` are carried into the report without filtering measurements or changing denominators. Compromised input is processed like other input; flags describe interpretation limits rather than validating the counts.

Usage

```
aggregate_burden(x, by = c("physical_slide_id", "organ"))
```

Arguments

x	Data frame containing grouping columns, physical_area_id, n_spots, n_profiles, red_area_mm2, and tissue_area_mm2. An optional character qc_flags column contains semicolon-separated flags; an empty string means no concern marked in the reviewed material, while NA means quality information is unavailable.
by	Column names defining summaries; defaults to physical slide and organ.

Value

One data-frame row per group, with area-weighted fractions and densities.

---

aggregate_event_burden

Aggregate event counts and relative positive area by slide

Description

Sum counts and union areas from disjoint physical measurement regions. The caller must resolve overlapping scans and event ownership before aggregation. Optional quality flags propagate without excluding compromised measurements.

Usage

```
aggregate_event_burden(x, by = c("physical_slide_id", "organ"))
```

Arguments

x	Data frame with grouping columns, physical_area_id, n_events, red_area_mm2 and tissue_area_mm2. Optional qc_flags are semicolon-separated character values; NA indicates unavailable quality information.
by	Grouping columns; defaults to physical slide and organ.

Value

Slide-level sums, area-weighted positive fraction and percentage, event density, and retained quality flags. No animal/group inference.

bs_analyze *Analyze using a configured worker*

Description

Analyze using a configured worker

Usage

```
bs_analyze(inputs, backend, out, parameters = list(), timeout = 3600)
```

Arguments

inputs	Named character vector of existing input files.
backend	A bs_backend object.
out	New output directory. It is committed only after validation.
parameters	Named list of JSON-serializable worker settings.
timeout	Maximum elapsed seconds for the worker.

Value

A validated bs_job receipt. Whole-slide capability depends on the worker.

bs_app *Explore slide results in a local Shiny application*

Description

Presents imported measurements without altering scientific values. CSV uploads contain a slide table; JSON uploads contain a result object with a `slides` table. Uploaded files are parsed as data, never evaluated as R code.

Usage

```
bs_app(
  results = NULL,
  launch = interactive(),
  images = NULL,
  title = "bloodspottR",
  group_by = "organ",
  group_label = "Tissue",
  slide_python = Sys.getenv("BLOODSPOTTR_PYTHON", "")
)
```

Arguments

results	A result list with a slides data frame, or NULL.
launch	Launch the application; FALSE returns a Shiny application.
images	Optional data frame or CSV manifest with image_id, slide_id, path, and optional label, annotations, scale_x, scale_y, offset_x, offset_y. Paths in CSV manifests are relative to the manifest. Images are PNG/JPEG previews or OpenSlide-supported native slides. Annotations are QuPath GeoJSON or point CSV with x/y image-pixel coordinates. Transforms map annotation coordinates into the displayed image: pixel * scale + offset.
title	Application title.
group_by	Metadata column used for grouping and filtering, default organ.
group_label	Display label for this grouping, default Tissue.
slide_python	Python executable with openslide-python and openslide-bin for native slides; unused for PNG/JPEG previews. No packages are installed.

Details

results can also be a saved result file path. The viewer's optional watch mode rereads this file and saved annotations every two seconds; failed reads retain the previous measurements. It never changes source files. Native QuPath project files and TissueGnostics acquisition projects need image/GeoJSON exports; they are not themselves OpenSlide image formats.

Value

A Shiny application object, invisibly when launched.

---

bs_as_cellspec *Exchange calibrated event candidates through cellspecR*

Description

Preserves candidate identity and class without asserting that an event is one cell. No intensity features, channel measurements or cell areas are invented. Coordinates must already be in the image frame in micrometres. A physical slide is mapped to cellspec's sample_id; no animal identity is inferred.

Usage

```
bs_as_cellspec(events, images, provenance = list())
```

Arguments

events	Data frame with event_id, image_id, physical_slide_id, x_um, y_um and event_type. Event IDs must be unique within an image.
images	Data frame with image_id, physical_slide_id, pixel_width_um, pixel_height_um, width_px and height_px. Every referenced image needs an explicit positive calibration; anisotropic pixels are supported.
provenance	Named list of additional provenance. The bloodspottR entry is reserved and records the candidate semantics and mapping.

Value

A validated cellspec object. Requires the optional cellspecR package.

---

bs_backend *Configure an external analysis worker*

Description

The explicitly trusted executable receives two final arguments: a JSON request path and an output staging directory. It must write result.json with schema_version 1, status 'complete', and a nonempty artifacts array of relative file paths. A subprocess is isolated and cleaned up after completion or timeout. Workers must keep helpers attached to their ancestry and join them before successful exit. Intentionally detached processes are unsupported. Logs are limited to 10 MiB per stream. This is not an OS security sandbox. No worker is installed or started by this constructor.

Usage

```
bs_backend(
  command,
  args = character(),
  version,
  operations = c("train", "analyze")
)
```

Arguments

command	Path or PATH name of a trusted executable.
args	Character arguments preceding the request/output paths.
version	Explicit worker version or source commit identifier.
operations	Supported operations, such as train or analyze.

Value

A bs_backend specification.

`bs_color_features` *Extract the study's three relative optical-density color features*

Description

Computes channel OD as `-log((channel + pseudocount) / (white + pseudocount))`. Output channels are green OD minus red OD, blue OD minus red OD, and mean channel OD, in that order. With the default settings this is the three-feature definition used by the study's RGB/OD networks, evaluated in double precision. The first two features are log red-to-green and red-to-blue chromaticity ratios. These are relative digital image features, not calibrated chromophore concentrations or fitted labels.

Usage

```
bs_color_features(rgb, white = 255, pseudocount = 1)
```

Arguments

`rgb`	Numeric matrix with three columns, or array with dimensions height by width by three channels. Channels must be ordered red, green, blue.
`white`	Finite positive common channel white reference, default 255.
`pseudocount`	Finite positive offset in input intensity units, default 1.

Details

No rescaling, clipping, white estimation or statistical fitting is performed. `white` is an explicit common channel reference in the input's intensity units. All pixels must be finite and within `[0, white]`; masks and missing pixels must be handled explicitly before calling this function.

Value

Numeric object with the same dimensions as `rgb`, with its final dimension named `od_g_minus_r`, `od_b_minus_r`, `od_mean`.

Examples

```
bs_color_features(rbind(c(255, 255, 255), c(255, 127, 63)))
```

bs_compare *Compare two runs with explicit support checks*

Description

common aligns shared physical slides and requires identical support_id, or identical legacy regions and acquired_pixels. This verifies declared support identity, not spatial registration. It never rescales to fabricate a common support. union retains unmatched slides with missing values and reports whether comparable support was documented.

Usage

```
bs_compare(runs, support = "common")
```

Arguments

runs	Named list of exactly two imported results or canonical tables.
support	"common" or "union".

Value

Data frame with baseline/run values, second-minus-first differences and support_verified. Run names and excluded IDs are attributes, preserved by bs_export_results() and bs_import_legacy() through comparison_context.

bs_cttir_inventory *Inspect a local CTTIR source suite without loading it*

Description

Reads immediate child directories with DESCRIPTION files. Duplicate package identities are retained and flagged: directory names are not package names. Source presence and declared exports do not establish runtime qualification. No source code, installation, services or network requests are executed.

Usage

```
bs_cttir_inventory(root, check_installed = FALSE)
```

Arguments

root	Existing suite directory.
check_installed	Inspect installed package metadata as well (no loading).

Value

Data frame of source paths, package names, versions, titles, declared literal exports, duplicate identity flags and optional installed versions. Malformed descriptions are retained with an error in `status`.

Examples

```
bs_cttir_inventory(tempdir())
```

---

bs_example *Deterministic demonstration results*

Description

Synthetic measurements for examples and application exploration, never study data or evidence of model performance.

Usage

```
bs_example()
```

Value

A result list with slides, groups and provenance.

---

bs_export_results *Export a frozen set of result tables*

Description

Creates a new directory with CSV tables, optional Excel output and a SHA-256 manifest. Existing destinations are never overwritten. CSV character values beginning with spreadsheet formula characters are escaped for safe viewing; canonical JSON preserves the original strings. Comparison run names and excluded slide IDs are saved as comparison_context and restored as comparison attributes by bs_import_legacy().

Usage

```
bs_export_results(results, out, xlsx = FALSE)
```

Arguments

`results`	A validated bloodspottR result object.
`out`	A new output directory.
`xlsx`	Include an Excel workbook (requires openxlsx2).

Value

The normalized delivery directory, invisibly.

`bs_import_legacy` *Import preserved result tables without changing source files*

Description

Reads a canonical CSV, the legacy physical-slide `slides.csv`, or saved `analysis.json` containing a `rows` table, or exported `results.json`. A folder must contain exactly one of these named files. The `p079` profile maps legacy area column names; it does not impose study-specific groups or thresholds. Saved QC/group metadata are preserved rather than inferred from signal. CSV identifiers are read as strings, including numeric-looking identifiers with leading zeros.

Usage

```
bs_import_legacy(path, profile = "p079")
```

Arguments

`path`	File or directory containing results.
`profile`	Adapter profile, `"p079"` or `"canonical"`.

Value

A `bs_result` object with canonical slide measurements, source SHA-256 identity, and available saved metadata. SHA-256 is an integrity receipt, not an authenticity guarantee.

`bs_profile_extent` *Compare a calibrated profile's diameter with an explicit reference*

Description

Measures one component from its filled-profile boundary coordinates. The maximum distance between any two supplied points equals the diameter of their convex hull. Rotating calipers evaluate that diameter without allocating an all-pairs distance matrix. Duplicated coordinates and collinear profiles are supported. Horizontal and vertical extents use the supplied image axes.

Usage

```
bs_profile_extent(coordinates_um, upper_diameter_um)
```

Arguments

`coordinates_um`	Numeric two-column matrix or data frame, x then y, describing one component in micrometres. At least one point is required.
`upper_diameter_um`	Finite positive upper single-profile reference in micrometres. There is no universal default.

Details

Coordinates must describe the outer profile, not only its annotation centre, and must already be calibrated in micrometres; apply separate x/y pixel calibration before use. Sparse boundary sampling can underestimate the true extent. The explicit upper reference requires slide-specific calibration. `single_compatible` only means that measured diameter does not exceed that reference. `exceeds_reference` is a review flag, never proof of multiple cells or an estimated cell count. Section plane, deformation, overlap and segmentation error remain unresolved by this measurement.

Value

One-row data frame with maximum diameter, x/y extents, reference, review flag and number of distinct supplied points. Equality to the upper reference is classified as single-compatible.

Examples

```
bs_profile_extent(rbind(c(0, 0), c(3, 0), c(3, 4), c(0, 4)), 5)
```

`bs_project` *Create or open a portable analysis project*

Description

Creation writes a versioned JSON manifest only to a new, empty directory. Opening never mutates files and does not start analysis services.

Usage

```
bs_project(path, create = FALSE)
```

Arguments

`path`	Project directory.
`create`	Whether to create a new project.

Value

A `bs_project` handle with its normalized root and manifest.

bs_report *Render a portable HTML results report*

Description

Writes a self-contained report and machine-readable tables to a new directory. This initial renderer reports supplied summaries and provenance; it does not invent training methods or embed unavailable original images. Print the HTML from a browser when a portable PDF is needed. The study-specific full-slide TeX renderer remains a separately qualified backend.

Usage

```
bs_report(
  results,
  out,
  title = "Histology burden report",
  author = "",
  background = "",
  xlsx = FALSE
)
```

Arguments

results	A bloodspottR result object.
out	A new delivery directory.
title	Report title.
author	Report author, explicitly supplied by the caller.
background	Curated background text, treated as plain text.
xlsx	Include the workbook.

Value

The report path, invisibly.

bs_review_confirm *Confirm an explicitly reviewed, saved annotation export*

Description

Saves a receipt only after reading and verifying the source image and the exported CSV. It does not click Save in QuPath or infer completeness from an empty file. A header-only CSV is accepted only with explicit scope assertions. Parent coordinates must already be supplied by the review client's adapter.

Usage

```
bs_review_confirm(plan, item_id, annotations, assertions)
```

Arguments

plan	A bs_review_plan object.
item_id	The next incomplete review item.
annotations	Saved CSV with class, x and y columns in parent coordinates.
assertions	Named list of ery, nuclei and vessel review scopes. Each is unreviewed, partial, complete or absent. At least one must be reviewed.

Value

A durable receipt, invisibly. Repeating an identical confirmation is idempotent.

---

bs_review_plan *Freeze a bounded image-review queue*

Description

A client-neutral queue for saved QuPath or other annotation exports. This function never launches or modifies a viewer. Image paths are explicit files; each item must have a unique ID and parent-image ID. Queue sampling preserves the caller's random-number state.

Usage

```
bs_review_plan(images, budget = 50L, seed = 1L, out)
```

Arguments

images	Data frame with item_id, parent_id and path.
budget	Maximum number of images to review.
seed	Nonnegative integer sampling seed.
out	New queue directory.

Value

A bs_review_plan object referring to the durable queue.

bs_run_backend *Run a validated external worker transaction*

Description

Run a validated external worker transaction

Usage

```
bs_run_backend(
  backend,
  operation,
  inputs,
  parameters = list(),
  out,
  timeout = 3600
)
```

Arguments

backend	A bs_backend object.
operation	A supported operation name.
inputs	Named character vector of existing input files.
parameters	Named list of JSON-serializable worker settings.
out	New output directory. It is committed only after validation.
timeout	Maximum elapsed seconds for the worker.

Value

A job receipt with status, backend, input hashes and output hashes.

bs_status *Inspect analysis state without launching workers*

Description

Inspect analysis state without launching workers

Usage

```
bs_status(x)
```

Arguments

x	A project or imported result object.

Value

A list describing schema, object kind and available results.

`bs_summarize` *Pool physical-slide measurements by declared groups*

Description

Rates divide summed numerators by summed tissue area, never average slide rates. Missing inputs propagate. Unknown group labels form explicit NA groups. Saved control labels are carried without interpreting biological identity.

Usage

```
bs_summarize(x, strata = NULL, qc_revision = NULL)
```

Arguments

`x`	Imported results or canonical physical-slide measurements.
`strata`	Character column names to group by. `NULL` gives one overall row.
`qc_revision`	Optional data frame keyed uniquely by `physical_slide_id`. It must cover exactly the measured slides and may replace metadata, but cannot replace measurements. The source object is not modified.

Value

Data frame with summed measurements, pooled rates, slide counts and a list column `slide_ids`. Only observed combinations are returned.

`bs_train` *Train using a configured worker*

Description

Train using a configured worker

Usage

```
bs_train(inputs, backend, out, parameters = list(), timeout = 3600)
```

Arguments

inputs	Named character vector of existing input files.
backend	A bs_backend object.
out	New output directory. It is committed only after validation.
parameters	Named list of JSON-serializable worker settings.
timeout	Maximum elapsed seconds for the worker.

Value

A validated bs_job receipt. Model accuracy is not established by this receipt.

---

bs_validate *Validate project or physical-slide measurement contracts*

Description

Missing measurements remain missing. Numeric counts must be whole, nonnegative doubles no larger than 2^53. Validation checks arithmetic and structure, not biological accuracy or geometric independence of slides.

Usage

```
bs_validate(x, level = "structure")
```

Arguments

x	A project, imported result, or canonical slide data frame.
level	Currently "structure" only.

Value

Invisibly TRUE, or an informative error.

bs_viewer_example *Create a synthetic annotated-slide viewer example*

Description

Writes eight small PNG images and GeoJSON annotation files to a temporary directory. Tissue labels span liver, spleen, lung and kidney; no study data is used. Pass the returned results and images to `bs_app()`. Delete `directory` when finished with the example.

Usage

```
bs_viewer_example()
```

Value

A list with results, images manifest and temporary directory.

Examples

```
demo <- bs_viewer_example()
if (interactive()) bs_app(demo$results, images = demo$images)
unlink(demo$directory, recursive = TRUE)
```

cell_pattern *Construct a cell point pattern inside observed tissue*

Description

Construct a cell point pattern inside observed tissue

Usage

```
cell_pattern(cells, window)
```

Arguments

`cells`	Data frame with unique `cell_id` and finite `x_um`, `y_um` columns.
`window`	A tissue observation window in micrometres from `tissue_window()`.

Value

A marked `spatstat.geom` point pattern. Points in holes or outside tissue, duplicate IDs and duplicate coordinates produce errors.

measure_burden *Calculate calibrated burden from classified pixels and objects*

Description

Inputs must already refer to the same evaluable tissue mask. This function checks units and arithmetic; it does not validate image segmentation. Zero evaluable area produces undefined densities, represented by NA.

Usage

```
measure_burden(
  n_spots,
  red_pixels,
  tissue_pixels,
  pixel_width_um,
  pixel_height_um,
  n_profiles = NULL
)
```

Arguments

n_spots	Nonnegative whole spot counts.
red_pixels	Nonnegative whole positive-pixel counts within tissue.
tissue_pixels	Nonnegative whole evaluable-tissue pixel counts.
pixel_width_um, pixel_height_um	Pixel dimensions in micrometres; either one value or one per measurement.
n_profiles	Optional nonnegative whole counts of resolved profiles. Omit when profile counts are unavailable or unsupported.

Value

A data frame of areas in square millimetres, counts, fractions and densities. The n_profiles and profiles_per_mm2 columns are NA when absent.

Examples

```
measure_burden(10, 200, 10000, 0.25, 0.25)
```

`measure_event_burden` *Measure positive events and relative positive area*

Description

An event is an object under the declared segmentation rule and need not be a single erythrocyte. Positive area is the union of red pixels within tissue, supplied independently of event masks. Splitting an event changes its count but must not change these red pixels. Quality flags are handled during aggregation and do not remove tissue or detections.

Usage

```
measure_event_burden(
  n_events,
  red_pixels,
  tissue_pixels,
  pixel_width_um,
  pixel_height_um
)
```

Arguments

`n_events`	Nonnegative whole event counts within tissue.
`red_pixels`	Nonnegative whole counts of union positive pixels in tissue.
`tissue_pixels`	Nonnegative whole tissue-pixel counts.
`pixel_width_um`, `pixel_height_um`	Pixel dimensions in micrometres.

Value

A data frame with event counts, calibrated areas, positive area fraction and percentage, and events per square millimetre. Zero tissue area yields undefined fractions and densities. This function validates arithmetic, not assay performance or single-cell identity.

Examples

```
measure_event_burden(2, 28, 64, 0.25, 0.25)
```

`tissue_window` *Convert local tissue polygons into a spatial observation window*

Description

Preserves holes and disconnected tissue. Input coordinates must be local image coordinates with no geographic CRS; the declared unit is explicit. Overlapping polygons are unioned before conversion.

Usage

```
tissue_window(polygons, coordinate_unit = c("um", "mm"))
```

Arguments

`polygons`	An `sf` or `sfc` object containing valid polygons only.
`coordinate_unit`	Either `um` or `mm`; returned coordinates use micrometres.

Value

A `spatstat.geom` polygonal observation window in micrometres.

Index

aggregate_burden, 2
aggregate_event_burden, 3

bs_analyze, 4
bs_app, 4
bs_as_cellspec, 5
bs_backend, 6
bs_color_features, 7
bs_compare, 8
bs_cttir_inventory, 8
bs_example, 9
bs_export_results, 9
bs_import_legacy, 10
bs_profile_extent, 10
bs_project, 11
bs_report, 12
bs_review_confirm, 12
bs_review_plan, 13
bs_run_backend, 14
bs_status, 14
bs_summarize, 15
bs_train, 15
bs_validate, 16
bs_viewer_example, 17

cell_pattern, 17

measure_burden, 18
measure_event_burden, 19

tissue_window, 20