Package: AiES (via r-universe)

November 2, 2026

Type Package

Title Axon Integrity Evaluation System for Microscopy Images

Version 1.0.0

Description Provides tools for the quantitative analysis of axon integrity in phase-contrast microscopy images. 'AiES' (Axon integrity Evaluation System) implements image pre-processing, adaptive thresholding, feature extraction, and support vector machine-based classification to compute indices such as the Axon Integrity Index (AII) and Degeneration Index (DI). Two built-in models are provided, and classifiers can be re-trained on user data. The package is designed for reproducible and automated analysis in neuroscience research.

License BSD_3_clause + file LICENSE

Copyright 2025 National Center of Neurology and Psychiatry

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

URL https://github.com/BreezyCave/AiES

BugReports https://github.com/BreezyCave/AiES/issues

Depends R (>= 3.6.0)

Imports EBImage, e1071, stats, fs, utils, data.table, rstudioapi, tiff

Suggests knitr, rmarkdown, tcltk, svDialogs

VignetteBuilder knitr

Config/pak/sysreqs cmake libfftw3-dev make libjpeg-dev libpng-dev libtiff-dev libuv1-dev

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

Date/Publication 2026-11-02 06:14:15 UTC

RemoteUrl https://github.com/breezycave/aies

RemoteRef HEAD

RemoteSha 2f9c9d0160fb673a57f2c915a5b53cb012634490

Contents

AiES-package 2
axDistmap 5
axQnt 9
axSvm 14

Index **19**

`AiES-package` *AiES: Axon Integrity Evaluation System*

Description

AiES quantifies axon integrity and degeneration from microscopy images. Axonal regions are segmented, described by shape, moment and Haralick texture features, and classified as intact or degenerate by a support vector machine. The package provides three functions:

- `axDistmap`: segments images and exports feature tables.
- `axSvm`: trains a classifier on your own feature tables.
- `axQnt`: applies a classifier and reports the Axon Integrity Index (AII) and Degeneration Index (DI).

Two built-in models are provided ("g0.1_APA" and "g0.2_ACC"). Whether they can be applied to your images depends on how those images were acquired and saved. This page describes which inputs are supported.

How AiES sees an image

Three properties of the input determine whether the segmentation behaves as it did for the images the built-in models were trained on:

1. **Polarity.** Axons are segmented as structures that are *darker* than their local surroundings, as they appear in phase-contrast images.
2. **Intensity range.** The first processing step is a contrast curve centered on half of the full intensity range. The training images were 8-bit and used the full 0-255 range. Images must therefore occupy the full range of their file format once read.
3. **Physical scale.** Every size-dependent parameter was defined at 0.791746 um per pixel. Images are resampled to this scale by default, which requires the pixel size of the input to be known.

Recommended input

The following inputs can be analyzed with the built-in models without further adjustment.

- **Imaging modality:** phase contrast, 20x objective, in which axons appear darker than the background. This is how the built-in models were trained.

- **File type:** TIFF (`.tif` / `.tiff`), uncompressed or losslessly compressed (e.g. LZW). Only TIFF files are searched for.
- **OME-TIFF (preferred):** files exported with OME-XML metadata (for example from Nikon NIS-Elements or Bio-Formats) carry both the pixel size and the bit depth of the camera, which are read automatically.
- **Channels:** a single grayscale channel.
- **Bit depth:** 8-bit data using the full 0-255 range, or higher bit depths that are either declared in OME-XML (OME-TIFF) or scaled to the full range of the file format by the acquisition software.
- **Pixel size:** known, and equal to or finer than 0.791746 um per pixel. Finer images are resampled down to this scale; coarser images would have to be interpolated up, which cannot restore detail that was never recorded.
- **Processing history:** raw acquisitions. No contrast adjustment, auto-levels, gamma correction, sharpening or denoising.

Supported with additional arguments

- **Pixel size:** set `pixelSize` to the physical size of one pixel in micrometers. This is required when the file does not store it, and recommended in general, because metadata formats differ between manufacturers. If the file also carries a pixel size, the value you specify is used and a warning reports both. It can be computed as the camera pixel pitch divided by the product of the objective and camera-adapter magnifications.
- **Color camera images (RGB):** set `colorMode = "rgb"`. The three channels are averaged into one grayscale image. This is the right choice for phase-contrast images recorded with a color camera.
- **Grayscale saved as RGB** (three identical channels): the default `colorMode = "gray"` uses the first channel, which loses nothing.

Applicable, but re-training is recommended

- **Other transmitted-light modalities** (bright field, DIC, other phase rings or objectives): the built-in models can be applied to quantify degeneration, but the contrast characteristics differ from the training images even at the same pixel size. Re-training with `axSvm` on your own data is expected to improve accuracy.

Requires re-training with axSvm

The following inputs can be processed, but the built-in models should not be applied to them. Train a classifier on your own data with `axSvm` instead.

- **Working at a finer scale than 0.791746 um per pixel** (`targetPixelSize` smaller than the default): size-dependent parameters are rescaled so that the segmentation stays sound, but perimeters, and therefore Circularity, grow as finer boundary detail is resolved, so features are not numerically comparable.

Not supported

- **Fluorescence and immunostaining images.** Axons are bright on a dark background in these images, the opposite polarity to phase contrast, and the preprocessing has not been validated for them. Their results should not be interpreted.

Not recommended

- **Multi-frame TIFF files** (z-stacks, time series): only the first frame is used.
- **Higher bit depth stored unscaled, without metadata** (e.g. 12-bit camera data saved as a 16-bit TIFF, maximum value 4095): the image is read as if it were very dark, and the segmentation changes completely. Export it as OME-TIFF, which records the significant bits, or scale it to the full range in the acquisition software. A warning is issued when an image uses less than 1/16 of its intensity range.
- **Lossy compression** (JPEG files, or TIFF files with JPEG compression): compression artifacts alter the texture features.
- **Processed images:** contrast-stretched, auto-leveled, gamma-corrected, sharpened or denoised images. The intensity distribution no longer matches the training data.
- **Annotations burned into the image:** scale bars, text or arrows are segmented as objects. Export images without overlays.
- **Images coarser than 0.791746 um per pixel** (e.g. low-magnification objectives): thin axons span too few pixels to be segmented reliably.
- **Stitched or tiled images** with visible seams or uneven illumination between tiles.
- **Z-stacks and time series in one file:** only the first frame is analyzed. Export each field of view as a separate file.
- **Binary masks:** detected automatically and skipped.

Quick reference

- 8-bit grayscale TIFF, pixel size stored in file: `axDistmap()`
- 8-bit grayscale TIFF, pixel size not stored: `axDistmap(pixelSize = ...)`
- OME-TIFF (12- or 16-bit): `axDistmap()` (pixel size and bit depth read automatically)
- 12-bit data saved unscaled as 16-bit TIFF without metadata: re-export as OME-TIFF, or scale to the full 16-bit range in the acquisition software
- RGB TIFF from a color camera: `axDistmap(pixelSize = ..., colorMode = "rgb")`

The same arguments are accepted by `axQnt` when `imprtImg = TRUE`.

Checking an image before analysis

The log printed for each file reports the image dimensions, number of channels, container bit depth, the source of the pixel size, and any rescaling applied. Inspecting the binary output of `axDistmap(binaryImg = TRUE)` for a few representative images is the most direct way to confirm that axons, and not the background between them, are being segmented.

Author(s)

Maintainer: Shinji Tokunaga <tokunaga@ncnp.go.jp> (ORCID)

Other contributors:

- Masabumi Funakoshi <funakoshi@ncnp.go.jp> [contributor]
- Toshiyuki Araki <taraki@ncnp.go.jp> (ORCID) [contributor]
- National Center of Neurology and Psychiatry [copyright holder]

See Also

axDistmap, axSvm, axQnt

---

axDistmap *Create distance map and binary image from image file*

Description

axDistmap processes image files (TIFF) to create distance maps and optionally binary images. It also computes various image features and exports them as a text file.

Images are resampled to a fixed physical scale (micrometers per pixel) rather than to a fixed pixel width, so that data acquired on different microscopes and cameras are directly comparable. Grayscale and RGB color phase-contrast images are supported via the colorMode argument.

Usage

```
axDistmap(
  pixelSize = NULL,
  targetPixelSize = 0.791746,
  resizeW = 900,
  subBack = 30,
  autoParam = TRUE,
  binaryImg = FALSE,
  featureSet = "standard",
  colorMode = "gray",
  imgType = "tiff",
  folder_paths = NULL,
  output_path = tempdir()
)
```

Arguments

pixelSize	Numeric or NULL. Physical size of one pixel of the *input* image, in micrometers. If NULL, the file metadata (OME-XML, TIFF resolution tags) is consulted; if those are absent, scale matching is disabled and resizeW is used instead. Metadata formats differ between manufacturers, so specifying pixelSize explicitly is recommended. If both are available, the specified value is used, and a warning reports both values. (default: NULL)

`targetPixelSize`
: Numeric or NULL. Pixel size, in micrometers, to resample the image to. The default is the scale at which the built-in models were trained (0.791746 um/px). To process the image without resizing it, set this parameter to the same value as pixelSize. Set to NULL to ignore physical scale entirely and fall back to `resizeW`. (default: 0.791746)

`resizeW`
: Numeric or NULL. Target width in pixels, used only when the physical scale cannot be resolved or when `targetPixelSize = NULL`. If NULL, the image is processed without resizing. (default: 900)

`subBack`
: Numeric. Connected components (objects) whose area is at most this many pixels at the reference scale (0.791746 um/px; 30 px = 18.8 um^2) are removed. With autoParam = TRUE, the value is converted automatically so that the same physical area is removed at any image resolution. (default: 30)

`autoParam`
: Logical. If TRUE, the pixel-unit segmentation parameters (`subBack` and the local threshold window) are rescaled so that they cover a constant physical size. The Haralick GLCM scales are always kept at 1 and 2 px. Has no effect when the physical scale is unknown. Set to FALSE only if `targetPixelSize` matches the reference scale, or if you need to reproduce the behavior of versions before 1.00.0 (together with `targetPixelSize = NULL`). (default: TRUE)

`binaryImg`
: Logical. If TRUE, exports binary image files. (default: FALSE)

`featureSet`
: Character. If "standard", exports the minimum set of features required by the built-in models of this package, plus Area. If "all", exports all computed features. (default: "standard").

`colorMode`
: Character. How the input image is reduced to a single grayscale channel:

  - "gray": single-channel input, or channels that all contain the same grayscale image (e.g. a grayscale image stored in an RGB container). Channel 1 is used.
  - "rgb": a true RGB color image, e.g. a phase-contrast image recorded with a color camera. The three channels are averaged.

  (default: "gray")

`imgType`
: Character. Output image format: "png", "jpg", or "tiff" (default: "tiff").

`folder_paths`
: Character vector. Folder paths to process. If NULL, a folder selection dialog will be shown. (default: NULL).

`output_path`
: Character. Output folder. If NULL, a folder selection dialog will be shown. (default: tempdir()).

Details

The function performs the following steps:

1. Reads the input image. Single-Channel (grayscale) images and 3-channel RGB images are supported
2. Reduces the image to a single grayscale channel according to `colorMode`
3. Resamples the image to `targetPixelSize` (or to `resizeW` if the scale is unknown or `targetPixelSize = NULL`)

4. Rescales the pixel-unit parameters if `autoParam = TRUE`
5. Applies contrast adjustment, filtering and local thresholding
6. Computes a distance map, then shape, Haralick, and moment features
7. Exports the processed images and the computed features

Value

This function doesn't return a value directly, but produces the following outputs:

- A distance map image file (format specified by `imgType`)
- A text file containing computed image features (named `<image>_<date>_ImageData.txt`)
- A parameter file (named `<folder>_<date>_Parameters.csv`)
- (Optional) A binary image file if `binaryImg = TRUE`

Input Images

Axons are segmented as structures darker than their local surroundings, as in phase-contrast images. Only TIFF files are processed. Images must occupy the full intensity range of their file format. Data from a camera with fewer bits than the TIFF container (e.g. 12-bit data in a 16-bit TIFF) should be exported as OME-TIFF, which records the significant bits (the actual bit depth of the data) and is rescaled automatically, or the data should be rescaled to use the full available range (e.g., 0-65535 for 16 bit images) within the acquisition software. A warning is issued for suspiciously dark images. The pixel size should be known (from OME-XML or TIFF metadata, or via `pixelSize`) to match images to the scale of the built-in models. If it cannot be determined, the image is resized to resizeW, and a warning is issued. Multi-frame TIFF files, lossy compression, contrast-adjusted images and burned-in scale bars should be avoided. See `AiES-package` (`?AiES`) for recommended and not recommended formats.

Physical Scale

Every pixel-unit parameter of the pipeline defines a physical size once the scale is fixed. At the reference scale of 0.791746 um/px, `subBack = 30` corresponds to 18.8 um^2, the local threshold window spans 13.5 x 13.5 um, and the Haralick GLCM offsets span 0.79 and 1.58 um. Resampling to the reference scale restores all of them simultaneously, which is what allows a classifier trained on one system to be applied to another.

If the `targetPixelSize` is set to a scale different from the reference scale, parameter rescaling keeps the physical size of the removed objects and of the local threshold window unchanged. However, the resulting feature values will not be numerically equivalent to those obtained at the reference scale. For example, since Haralick features are computed at fixed pixel scales, they will sample texture at a finer physical distance. Therefore, instead of using a built-in model trained at the reference scale, it is recommended to retrain the classifier using `axSvm`.

Color Handling

Depending on how a TIFF file was exported, the same field of view may be stored as a 2D grayscale array, a 3-channel RGB image, or a multi-frame stack. `axDistmap` reads every channel or frame, reports the detected structure, and then reduces the image to a single channel according to `colorMode`. "gray" (default) uses the first channel, while "rgb" averages the three channels. If "rgb" is selected

but the image does not have exactly three channels, the file is skipped with a warning. Multi-frame files (z-stacks, time series) are not recommended, as only the first frame is analyzed.

Feature Export

All features are exported under short names (see section "Feature names"), which are the names used by the built-in models. If featureSet = "standard", the predictors of the built-in models (Per, Ras, Ran, ecc, mjx, ENT, SVA, SV2, CON and cir) and Area (the area weight for AII/DI) are exported. The 'cir' feature represents Circularity and is calculated as: cir = (Area * pi * 4) / (Per^2)

If featureSet = "all", all computed features from EBImage's computeFeatures functions (shape, moment, and haralick) plus 'cir' are exported.

Parameter file

For every output folder a file `<folder>_<date>_Parameters.csv` is written next to the feature files, with one row per processed image: `FileName` (path relative to the input folder), `DataFile` (the matching ImageData.txt), `Date`, `AiESVersion`, `WidthPx`, `HeightPx`, `Channels`, `ContainerBits`, `SignificantBits`, `BitRescaled`, `ColorMode`, `ArgPixelSize` (the `pixelSize` argument), `MetaPixelSize` and `MetaSource` (value found in the file and where), `PixelSize` (effective scale, um/px, of the analyzed image; NA when unknown), `ScaleMode` (`scale-matched`, `fixed-width` or `no-resize`), `ResizeW` (working width, px), `SubBack` and `ThreshWH` (values actually used) and `FeatureSet`. `axSvm` reads `PixelSize` from this file to record the training scale with the model. The feature files themselves contain features only.

Feature names

Each entry reads: short name = original EBImage column name (cir is calculated by AiES).

- Shape: Area = s.area (area weight for AII / DI), Per = s.perimeter, Ram = s.radius.mean, Ras = s.radius.sd, Ran = s.radius.min, Rax = s.radius.max.
- Moment: mcx = m.cx, mcy = m.cy, mjx = m.majoraxis, ecc = m.eccentricity, the = m.theta.
- Haralick, scale 1: ASM = h.asm.s1, CON = h.con.s1, COR = h.cor.s1, VAR = h.var.s1, IDM = h.idm.s1, SAV = h.sav.s1, SVA = h.sva.s1, SEN = h.sen.s1, ENT = h.ent.s1, DVA = h.dva.s1, DEN = h.den.s1, F12 = h.f12.s1, F13 = h.f13.s1.
- Haralick, scale 2: SM2 = h.asm.s2, ON2 = h.con.s2, OR2 = h.cor.s2, AR2 = h.var.s2, DM2 = h.idm.s2, AV2 = h.sav.s2, SV2 = h.sva.s2, SE2 = h.sen.s2, NT2 = h.ent.s2, VA2 = h.dva.s2, EN2 = h.den.s2, F22 = h.f12.s2, F32 = h.f13.s2.
- Circularity: cir.

Note

- Output files are saved under `output_path` in a dated subfolder.
- Feature columns are exported under the short names above. Text files written by earlier versions (original EBImage names) are still accepted by `axSvm` and `axQnt` and converted automatically.
- Haralick features are computed at scales of 1 and 2 px (scale 1: ASM, CON, ...; scale 2: SM2, ON2, ...) at any resolution.

Examples

```
# Basic usage with default parameters
# NOTE: This example requires a GUI environment for interactive folder selection.
if (interactive()){
axDistmap()
}


# Declare the scale of the input images explicitly
# (the sample images are already resized to 900 px width, i.e. 0.791746 um/px)
img_dir <- system.file("extdata", "Degenerate_Images", package = "AiES")
axDistmap(pixelSize = 0.791746, folder_paths = img_dir, output_path = tempdir())

# Work at the input's own scale (parameters are rescaled automatically; for the
# sample images this equals the reference scale)
axDistmap(pixelSize = 0.791746, targetPixelSize = 0.791746,
folder_paths = img_dir, output_path = tempdir())

# Average the three channels (the sample images store a grayscale image in an RGB container)
axDistmap(pixelSize = 0.791746, colorMode = "rgb",
folder_paths = img_dir, output_path = tempdir())

# Legacy behavior: ignore physical scale, resize to a fixed width
axDistmap(targetPixelSize = NULL, resizeW = 900, featureSet = "all",
folder_paths = img_dir, output_path = tempdir())


# Process with a custom width and background threshold
img_dir <- system.file("extdata", "Degenerate_Images", package = "AiES")
axDistmap(subBack = 20, targetPixelSize = NULL, resizeW = 300, imgType = "png",
folder_paths = img_dir, output_path = tempdir())
```

`axQnt` *Quantify Axon Integrity and Degeneration Indices from Images or Feature Data*

Description

`axQnt` calculates the Axon Integrity Index (AII) and Degeneration Index (DI) from axon image files (.tiff) or from text files (.txt) of image features exported by `axDistmap`. The function uses a pre-trained SVM model to classify axonal regions and computes indices based on classified areas. `axQnt` supports both interactive GUI-based file/folder selection and direct path specification for input and output, with cross-platform compatibility.

Either one of the built-in models (`model = "g0.1_APA"` or `"g0.2_ACC"`) or any model produced by `axSvm` can be used.

Usage

```
axQnt(
  model = "g0.1_APA",
  imprtImg = TRUE,
  subBack = 30,
  pixelSize = NULL,
  targetPixelSize = NULL,
  resizeW = 900,
  autoParam = TRUE,
  colorMode = "gray",
  binaryImg = FALSE,
  distMapImg = FALSE,
  expSip = TRUE,
  svm_model_path = NULL,
  input_dirs = NULL,
  output_dir = tempdir()
)
```

Arguments

`model`	Character or NULL. Built-in model to apply: "g0.1_APA" or "g0.2_ACC". Ignored when `svm_model_path` is given. If both `model` and `svm_model_path` are NULL, a file selection dialog is shown to choose one model file. (default: "g0.1_APA")
`imprtImg`	Logical. If TRUE, imports and processes image files (.tiff). If FALSE, uses precomputed feature data (.txt). (default: TRUE) The image-processing arguments (`subBack`, `pixelSize`, `targetPixelSize`, `resizeW`, `autoParam`, `colorMode`, `binaryImg` and `distMapImg`) are used only when `imprtImg = TRUE`. Arguments shared with `axDistmap` are described identically in both functions.
`subBack`	Numeric. Connected components (objects) whose area is at most this many pixels at the reference scale (0.791746 um/px; 30 px = 18.8 um^2) are removed. With autoParam = TRUE, the value is converted automatically so that the same physical area is removed at any image resolution. (default: 30)
`pixelSize`	Numeric or NULL. Physical size of one pixel of the *input* image, in micrometers. If NULL, the file metadata (OME-XML, TIFF resolution tags) is consulted; if those are absent, scale matching is disabled and `resizeW` is used instead. Metadata formats differ between manufacturers, so specifying `pixelSize` explicitly is recommended. If both are available, the specified value is used, and a warning reports both values. (default: NULL)
`targetPixelSize`	Numeric or NULL. Pixel size, in micrometers, to resample the image to. The default is the scale recorded with the model (0.791746 um/px for the built-in models, and for models without a recorded scale), so that images are brought to the scale the model was trained at. To process the image without resizing it, set this parameter to the same value as pixelSize. Set to NULL to ignore physical scale entirely and fall back to `resizeW`. (default: taken from the model; leave the argument out to use it)

`resizeW` Numeric or NULL. Target width in pixels, used only when the physical scale cannot be resolved or when `targetPixelSize = NULL`. If NULL, the image is processed without resizing. (default: 900)

`autoParam` Logical. If TRUE, the pixel-unit segmentation parameters (`subBack` and the local threshold window) are rescaled so that they cover a constant physical size. The Haralick GLCM scales are always kept at 1 and 2 px. Has no effect when the physical scale is unknown. Set to FALSE only if `targetPixelSize` matches the reference scale, or if you need to reproduce the behavior of versions before 1.00.0 (together with `targetPixelSize = NULL`). (default: TRUE)

`colorMode` Character. How the input image is reduced to a single grayscale channel:

- "gray": single-channel input, or channels that all contain the same grayscale image (e.g. a grayscale image stored in an RGB container). Channel 1 is used.
- "rgb": a true RGB color image, e.g. a phase-contrast image recorded with a color camera. The three channels are averaged.

(default: "gray")

`binaryImg` Logical. If TRUE, exports binary image files. (default: FALSE) In `axQnt`, the files are written as TIFF.

`distMapImg` Logical. If TRUE, exports the distance map image of each file (the same image as written by `axDistmap`), as a TIFF file. Used with `imprtImg = TRUE`. (default: FALSE)

`expSip` Logical. If TRUE, exports single image prediction results as .csv files. (default: TRUE)

`svm_model_path` Character or NULL. Path to one user-supplied SVM model file (.RData or .svm), typically produced by `axSvm` (not a folder). The file may contain other objects (e.g. a saved workspace); the first object of class "svm" is used, under any name, and the other objects do not affect the analysis. If given, it takes precedence over `model`. (default: NULL)

`input_dirs` Character. Path to the folders containing input files (.tiff or .txt). If NULL, a folder selection dialog will be shown. (default: NULL)

`output_dir` Character. Path to the output directory. If NULL, a folder selection dialog will be shown; if empty, current working directory is used. One subfolder per input folder is created below it (see section "Output Folders"). (default: tempdir())

Details

The function implements a complete quantification pipeline:

1. **Model loading**: Loads a built-in model, or a pre-trained SVM model from the specified file or via GUI selection. Automatically detects SVM objects within the loaded file.
2. **Predictor detection**: Reads the predictors the model was trained on directly from the model object, and maps them back to the corresponding `axDistmap` column names. Any model produced by `axSvm` therefore works without further configuration.
3. **Input handling**: If input paths are not specified, the user is prompted to select folders via a GUI dialog.

4. **Data preprocessing**: Depending on `imprtImg`, either processes image files (.tiff) using exactly the same pipeline as `axDistmap` or directly utilizes precomputed feature data (.txt).
5. **SVM classification**: Classifies axonal regions into "Degenerate" or "Intact".
6. **Index calculation**: Calculates Axon Integrity Index (AII) and Degeneration Index (DI) based on classified areas:

$$AII = \frac{\text{Area of Intact Axons}}{\text{Total Axonal Area}}$$

$$DI = \frac{\text{Area of Degenerate Axons}}{\text{Total Axonal Area}}$$

7. **Result saving**: Exports summary and optional detailed prediction files with unique filenames to prevent overwriting.

Value

Invisibly returns the combined summary table (NULL if no file could be processed) and generates the following outputs:

- `Summary_<date>.csv` in `output_dir`: the calculated Axon Integrity Index (AII) and Degeneration Index (DI) for every file of every input folder. `FileName` is the path relative to the input folder. Not written if no file could be processed.
- In the subfolder of each input folder: `<folder>_<date>_Summary.csv` (the same table for that folder only).
- (Optional) Single image prediction results as .csv files if `expSip = TRUE`.
- (Optional) Binary images if `binaryImg = TRUE`, and distance map images (`<file>_<date>_DistMap.tiff`) if `distMapImg = TRUE`.
- (Image route only) In the same subfolder, `<folder>_<date>_Parameters.csv`: one row per processed image, with the same columns as the parameter file of `axDistmap` (see there). `FileName` is the path relative to the input folder; `DataFile` is the single image prediction file, or empty if `expSip = FALSE`; `FeatureSet` is "all".

Input Images

Axons are segmented as structures darker than their local surroundings, as in phase-contrast images. Only TIFF files are processed. Images must occupy the full intensity range of their file format. Data from a camera with fewer bits than the TIFF container (e.g. 12-bit data in a 16-bit TIFF) should be exported as OME-TIFF, which records the significant bits and is rescaled automatically, or be scaled to the full range in the acquisition software. A warning is issued for suspiciously dark images. The pixel size must be known (from OME-XML or TIFF metadata, or via `pixelSize`) for the built-in models to apply. Lossy compression, contrast-adjusted images and burned-in scale bars should be avoided. See `AiES-package` (`?AiES`) for recommended and not recommended formats.

Physical Scale

The built-in models were trained on images analyzed at 0.791746 um/px. By default, `axQnt` resamples every input image to match the model's scale. This keeps pixel-based pipeline parameters such as background area threshold, local threshold window, and Haralick GLCM offsets consistent with the physical distances used during training.

If an image's scale differs from that of the model (for example, when pixel size is unknown and the image is resized with `resizeW`, or when `targetPixelSize` is set), a warning at the end summarizes the number of such images, their working scales, and the ratio to the model's scale. The working scale for each image is also saved in the `PixelSize` column.

Features such as Perimeter, Circularity, and the Haralick texture features are sensitive to image resolution. Changes in pixel size affect these values, and simply rescaling does not fully correct for this. For accurate results, create a new model with `axSvm` using data acquired at the new resolution.

Output Folders

For each input folder, a subfolder `<folder>_<date>_output_files` is created in `output_dir` (the same naming as `axDistmap`). If several input folders have the same name, they are numbered (`<folder>_2_<date>_output_files`, ...). The single image prediction files, binary images, the folder's summary and its parameter file are written there. The combined `Summary_<date>.csv` of all input folders is written directly to `output_dir`.

File Naming and Overwrite Policy

- Output files are automatically named with timestamps in ISO 8601 format (YYYY-MM-DD).
- If a file name already exists, a unique name with a numeric suffix is generated to avoid overwriting.
- If the output directory does not exist, it will be created automatically.

GUI Support

- RStudio (rstudioapi), tcltk, and svDialogs are supported for file/folder selection.
- If no GUI is available, the user is prompted to enter the path manually.
- Cross-platform compatibility for Windows, macOS, and Linux.

Note

- Input .txt files are expected to carry the short feature names exported by `axDistmap`. Files written by earlier versions (original EBImage names) are converted automatically.
- The column Area must be present, as it provides the area weighting for AII and DI.
- Output files include timestamp in ISO 8601 format (YYYY-MM-DD).

Examples

```
# Interactive mode: process .tiff images with GUI dialogs
# NOTE: This example requires a GUI environment for interactive folder selection.
if (interactive()){
axQnt(imprtImg = TRUE, expSip = TRUE)
}


# Apply the built-in g0.1 APA model to package-included images
img_dir <- system.file("extdata", "Degenerate_Images", package = "AiES")
axQnt(model = "g0.1_APA", imprtImg = TRUE, input_dirs = img_dir, output_dir = tempdir())
```

```
# Declare the scale of the input images explicitly
axQnt(model = "g0.1_APA", pixelSize = 0.791746, imprtImg = TRUE,
      input_dirs = img_dir, output_dir = tempdir())

# Apply the built-in g0.2 ACC model instead
axQnt(model = "g0.2_ACC", imprtImg = TRUE, input_dirs = img_dir, output_dir = tempdir())

# Apply a model re-trained with axSvm()
my_model <- file.path(tempdir(), "my_classifier.svm")
if (file.exists(my_model)) {
  axQnt(svm_model_path = my_model, imprtImg = TRUE,
        input_dirs = img_dir, output_dir = tempdir())
}


# Utilize package-included precomputed feature data from .txt files
txt_dir <- system.file("extdata", "Degenerate_txt", package = "AiES")
axQnt(model = "g0.1_APA", imprtImg = FALSE, input_dirs = txt_dir, output_dir = tempdir())
```

---

axSvm *Build SVM Classifier for Axon Image Feature Classification*

Description

Trains a Support Vector Machine (SVM) classifier to distinguish between degenerative and intact axon states using image feature data. Supports both direct-path input and interactive GUI selection.

The feature set can either follow one of the built-in models (`model = "g0.1_APA"` or `"g0.2_ACC"`) or be specified freely (`model = "custom"`), which allows re-training on data acquired under a different imaging system or resolution.

Usage

```
axSvm(
  model = "g0.1_APA",
  features = NULL,
  targetPixelSize = NULL,
  nCst = NULL,
  nGmm = NULL,
  nCrss = 5,
  maxNumImg = 12,
  maxNumObj = 1000,
  seed = 1,
  degenerate_path = NULL,
  intact_path = NULL,
  output_data_path = tempdir(),
  output_model_path = tempdir()
)
```

Arguments

model	Character. Feature set to train on: "g0.1_APA", "g0.2_ACC", or "custom". For the two built-in sets, the hyperparameters of the built-in model are used unless nCst / nGmm are given explicitly. (default: "g0.1_APA")
features	Character vector. Features to use when model = "custom". Names may be given either as the short names used by the built-in models (e.g. "Ras", "ENT") or as the original EBImage column names (e.g. "s.radius.sd", "h.ent.s1"). If NULL, all available feature columns are used, except Area (the area weight), the centroid and orientation features (mcx, mcy, the), and DVA and VA2, which duplicate CON and ON2. Ignored unless model = "custom". (default: NULL)
targetPixelSize	Numeric or NULL. Scale, in micrometers per pixel, at which the training images were analyzed (the effective scale after resampling). It is stored with the model, and axQnt uses it as the default working scale and warns when an image is analyzed at a different scale. If NULL and the input data have a *_Parameters.csv file written by axDistmap in the same folder, its PixelSize is used when every input file is found there and all share one value. Otherwise, a message gives the reason and no scale is recorded. (default: NULL)
nCst	Numeric or NULL. SVM cost parameter (C-value) controlling margin hardness. Higher values increase model complexity. If NULL, the cost of the built-in model selected by model is used (1.189 for "g0.1_APA", 1.110 for "g0.2_ACC"), or 1 (the default of e1071::svm) for model = "custom". (default: NULL)
nGmm	Numeric or NULL. SVM gamma parameter (γ) controlling the RBF kernel width. Smaller values mean a larger kernel radius. If NULL, the gamma of the built-in model selected by model is used (0.0871 for "g0.1_APA", 0.207 for "g0.2_ACC"), or 0.1 for model = "custom". (default: NULL)
nCrss	Integer. Number of folds for cross-validation. Recommended values 5-10. (default: 5)
maxNumImg	Integer or NULL. Maximum number of images (feature files) to be used from each group (Degenerate and Intact). If the number of available files in a group exceeds this value, a subset of files of size maxNumImg is selected randomly according to a uniform probability distribution (see seed and section "Sampling"). If NULL, all available files are used. The default, 12, corresponds to the number of images per group used for training the built-in models. (default: 12)
maxNumObj	Integer or NULL. Maximum number of objects to be used from each image. If the number of available objects in an image exceeds this value, a subset of objects of size maxNumObj is selected randomly according to a uniform probability distribution (see seed). Because the number of objects per image varies with the degree of degeneration and/or axon density, this limit prevents a small number of images with many objects from dominating the training data. If NULL, all available objects are used. The default, 1000, corresponds to the value used for training the built-in models. (default: 1000)
seed	Integer or NULL. Random seed for image and object sampling, ensuring that identical input yields identical training data. The R session's random number

state is restored upon function exit. Specify a different value to assess the robustness of the model with respect to sampling variability. If NULL, no seed is set and the function uses the current random number stream. Call `set.seed()` beforehand for reproducibility. (default: 1)

`degenerate_path`
: Character. Feature data for the "Degenerate" group: the path to one folder containing .txt files, or a character vector of .txt file paths. Only one folder can be specified. If NULL, a GUI folder selection dialog will be shown. (default: NULL)

`intact_path`
: Character. Feature data for the "Intact" group: the path to one folder containing .txt files, or a character vector of .txt file paths. Only one folder can be specified. If NULL, a GUI folder selection dialog will be shown. (default: NULL)

`output_data_path`
: Character or NULL. Path to save combined feature data (TSV). If NULL, a GUI save dialog will be shown. If a directory is specified, the file will be saved there with an autogenerated name. If a file name already exists, a unique name will be generated. If the directory does not exist, a warning is issued and the file is saved in the current working directory under the autogenerated name. (default: tempdir())

`output_model_path`
: Character or NULL. Path to save trained SVM model (Rdata format). If NULL, a GUI save dialog will be shown. If a directory is specified, the file will be saved there with an autogenerated name. If a file name already exists, a unique name will be generated. If the directory does not exist, a warning is issued and the file is saved in the current working directory under the autogenerated name. (default: tempdir())

Details

The function implements a complete machine learning pipeline:

1. **Input handling**: If no input paths are specified, a GUI dialog will prompt the user to select the appropriate folders. The dialog is compatible with RStudio, tcltk, and svDialogs, ensuring cross-platform support.
2. **Data preprocessing**: The function reads .txt files from the specified folders, using up to `maxNumImg` files per group and up to `maxNumObj` objects per file. It then merges the extracted data and assigns group labels ("Degenerate" or "Intact") to each entry. Columns that are not present in every input file (e.g. files exported with different `featureSet` settings of `axDistmap`) are reported and not used.
3. **Name conversion**: `axDistmap` exports short feature names used by the built-in models (e.g., Ras, ENT). If columns still use the original EBImage names, these are converted to the corresponding short names (e.g., s.radius.sd -> Ras) prior to training.
4. **Feature selection**: Feature selection is restricted to those specified in `model` or `features`.
5. **Model training and evaluation**: Model training and evaluation are performed using the e1071 package with a radial basis function (RBF) kernel. Cross-validation is applied.
6. **Result saving**: The merged feature data and the trained SVM model are exported to disk. Unique filenames are generated automatically to prevent overwriting.

Value

Invisibly returns the trained SVM model object and produces the following outputs:

- Combined feature data file for machine learning (TSV).
- Trained SVM model file (Rdata format)

Sampling

With an integer seed, the random number generator is initialized using `set.seed(seed, kind = "Mersenne-Twister", normal.kind = "Inversion", sample.kind = "Rejection")` prior to sampling. This ensures that identical input always produces identical training data, and the original random number state of the R session is restored upon function exit. Before sampling, files within each group are sorted in a locale-independent manner, guaranteeing that file selection is consistent across operating systems. Sampling proceeds by selecting images first (Degenerate group, then Intact group), followed by objects within each file. Within each file, objects retain their original order. For each group, the numbers of available and selected images and objects are reported.

GUI Support

- RStudio (rstudioapi), tcltk and svDialogs are supported for file/folder selection.
- If no GUI is available, the input paths must be given as arguments; output paths can be entered manually.

Built-in feature sets

- **g0.1 APA** (5 features): Ras, ENT, Per, SVA, SV2; gamma = 0.0871, cost = 1.189. Recommended primary model.
- **g0.2 ACC** (6 features): CON, ecc, mjx, cir, Ras, Ran; gamma = 0.207, cost = 1.110.

Note

- Compatible with feature data from `axDistmap`
- Default output file names include a timestamp in ISO 8601 format (YYYY-MM-DD)
- Model files can be reloaded using base::load()
- Models produced here can be applied directly with `axQnt`, which reads the required predictors from the model object itself.

Examples

```
# Interactive mode with GUI prompts
# NOTE: This example requires a GUI environment for interactive folder selection.
if (interactive()){
axSvm()
}


# Re-train the g0.1 APA feature set on your own data
deg_dir <- system.file("extdata", "Degenerate_txt", package = "AiES")
```

```
int_dir <- system.file("extdata", "Intact_txt", package = "AiES")
axSvm(model = "g0.1_APA", degenerate_path = deg_dir, intact_path = int_dir,
      output_data_path = tempdir(), output_model_path = tempdir())

# Train on a freely chosen feature set with custom hyperparameters
axSvm(model = "custom", features = c("Ras", "ENT", "cir"),
      nCst = 5, nGmm = 0.05, nCrss = 10,
      degenerate_path = deg_dir, intact_path = int_dir,
      output_data_path = tempdir(), output_model_path = tempdir())
```

Index

AiES *(*AiES-package*)*, 2
AiES-package, 2
axDistmap, *2*, 5, *5*, *9–13*, *15–17*
axQnt, *2*, *4, 5*, *8*, 9, *15*, *17*
axSvm, *2, 3*, *5*, *7–9*, *11*, *13*, 14