Package: AiES (via r-universe)

June 2, 2026

Type Package

Title Axon Integrity Evaluation System for Microscopy Images

Version 0.99.6

Maintainer Shinji Tokunaga <tokunaga@ncnp.go.jp>

Description Provides tools for the quantitative analysis of axon integrity in microscopy images. It 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). 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 7.3.2

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

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

Imports EBImage, e1071, stats, fs, data.table

Suggests utils, stringr, colorspace, dplyr, knitr, tcltk, rstudioapi, 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 2025-06-04 15:51:41 UTC

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

RemoteRef HEAD

RemoteSha 66296a915527d8b6cf86a2b15d024c34f3f26482

Contents

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

Description

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

Usage

```
axDistmap(
  subBack = 30,
  resizeW = 900,
  binaryImg = FALSE,
  allFeatures = FALSE,
  imgType = "tiff",
  folder_paths = NULL,
  output_path = tempdir()
)
```

Arguments

`subBack`	Numeric. Area threshold in pixels: connected components (objects) with area less than or equal to this value will be removed as background. (default: 30).
`resizeW`	Numeric. Target width in pixels for resizing each imported image (default: 900, recommended > 700). Larger values preserve more image detail and may improve analysis accuracy, but also increase memory usage and computation time. Note that while higher resolutions can enhance result quality, beyond a certain point, further increasing the image size yields diminishing returns in accuracy but continues to increase computational cost.
`binaryImg`	Logical. If TRUE, exports binary image files (default: FALSE).
`allFeatures`	Logical. If TRUE, exports data of all computed features (default: FALSE).
`imgType`	Character. Output image format: "png", "jpg", or "tiff" (default: "tiff").
`folder_paths`	Character vector. A character vector of folder paths to process. If NULL, a folder selection dialog will be shown. (default: NULL).
`output_path`	Character vector. A character vector of folder paths to process. If NULL, a folder selection dialog will be shown. (default: NULL).

Details

The function performs the following steps:

1. Reads and resizes the input TIFF image
2. Applies various image processing techniques (contrast adjustment, filtering, thresholding)
3. Computes a distance map
4. Computes shape, Haralick, and moment features
5. Exports the processed images as image files
6. Exports the computed features as a tab-separated text file

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 "'original_filename'_'current_date'_ImageData.txt")
- (Optional) A binary image file if `binaryImg` = `TRUE` (format specified by `imgType`)

Feature Export

The function always exports computed features as a text file. If allFeatures = FALSE, it exports a subset of features including:

- s.area (EBImage::computeFeatures.shape)
- m.eccentricity (EBImage::computeFeatures.moment)
- s.radius.sd (EBImage::computeFeatures.shape)
- h.sva.s2 (EBImage::computeFeatures.haralick)
- h.idm.s1 (EBImage::computeFeatures.haralick)
- h.sen.s1 (EBImage::computeFeatures.haralick)
- m.majoraxis (EBImage::computeFeatures.moment)

If allFeatures = TRUE, it exports all computed features from EBImage's computeFeatures functions (shape, moment, and haralick) plus an additional 'Cir' feature. The 'Cir' feature represents Circularity and is calculated as: Cir = (s.area * pi * 4) / (s.perimeter^2), where s.area and s.perimeter are from EBImage::computeFeatures.shape.

Note

- The function will prompt the user to select a directory containing TIFF files.
- It processes all TIFF files in the selected directory.
- Once there are no unprocessed files left in the selected directory, the function will prompt the user to choose whether to process another folder.
- Processing will continue until the user cancels the operation.
- Output files (images and feature data) are saved in the same directory as the input files.

- The feature data text file is named using the format: "'original_filename'_'current_date'_ImageData.txt"
- If allFeatures = FALSE, only features required for the support vector machine learning in this package are exported.
- If allFeatures = TRUE, all features from EBImage package plus Circularity are exported.

Examples

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

# Create binary images and export all EBImage features plus Circularity as PNG

img_dir <- system.file("extdata", "Degenerate_Images", package = "AiES")
axDistmap(subBack = 50, binaryImg = TRUE, allFeatures = TRUE, imgType = "png",
folder_paths = img_dir, output_path = tempdir())
# Export all EBImage features plus Circularity without creating binary images
img_dir <- system.file("extdata", "Degenerate_Images", package = "AiES")
axDistmap(binaryImg = FALSE, allFeatures = TRUE,folder_paths = img_dir, output_path = tempdir())


# Process images and export as TIFF without binary images
# Process with custom image resize and background threshold
# Only exports features needed for SVM learning
img_dir <- system.file("extdata", "Degenerate_Images", package = "AiES")
axDistmap(subBack = 20, resizeW = 300, binaryImg = FALSE, allFeatures = FALSE, 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 precomputed feature data (.txt). 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.

Usage

```
axQnt(
  imprtImg = TRUE,
  subBack = 30,
  resizeW = 900,
  binaryImg = FALSE,
  expSip = TRUE,
  svm_model_path = tempdir(),
  input_dirs = NULL,
  output_dir = tempdir()
)
```

Arguments

`imprtImg`	Logical. If TRUE, imports and processes image files (.tiff). If FALSE, uses precomputed feature data (.txt). (default: TRUE)
`subBack`	Numeric. Area threshold in pixels: connected components (objects) with area less than or equal to this value will be removed as background. (default: 30)
`resizeW`	Numeric. Target width in pixels for resizing each imported image (default: 900). Larger values may improve analysis accuracy up to a point, but also increase memory and computation time. Excessively large values may not further improve results.
`binaryImg`	Logical. If TRUE, exports binary image files. (default: FALSE)
`expSip`	Logical. If TRUE, exports single image prediction results as .csv files. (default: TRUE)
`svm_model_path`	Character. Path to a pre-trained SVM model file (.RData or .svm). If NULL, a file selection dialog will be shown. (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. (default: NULL)

Details

The function implements a complete quantification pipeline:

1. **Model loading**: Loads a pre-trained SVM model from the specified file or via GUI selection. Automatically detects SVM objects within the loaded file.
2. **Input handling**: If input paths are not specified, the user is prompted to select folders via a GUI dialog (cross-platform support for RStudio, tcltk, svDialogs, or manual input).
3. **Data preprocessing**: Depending on `imprtImg`, either processes image files (.tiff) to extract features utilizing the same pipeline as `axDistmap` or directly utilizes precomputed feature data (.txt).
4. **SVM classification**: Utilizes the loaded SVM model to classify axonal regions into "Degenerate" or "Intact" based on morphological features.

5. **Index calculation**: Calculates Axon Integrity Index (AII) and Degeneration Index (DI) based on classified areas:

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

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

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

Value

The function does not return values directly but generates the following outputs:

- A summary .csv file containing the calculated Axon Integrity Index (AII) and Degeneration Index (DI) for each file.
- (Optional) Single image prediction results as .csv files if `expSip = TRUE`.
- (Optional) Processed image data as .txt files if `imprtImg = TRUE`.

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

- The function requires a pre-trained SVM model compatible with the feature set used by `axDistmap` and `axSvm`.
- Required features for SVM classification: m.eccentricity, s.radius.sd, h.sva.s2, h.idm.s1, h.sen.s1, m.majoraxis.
- When `imprtImg = TRUE`, images are processed using the same pipeline as `axDistmap` with the specified `resizeW` and `subBack` parameters.
- Single image prediction results are exported only if `expSip = TRUE`.
- Compatible with feature data and models from `axDistmap` and `axSvm`.
- 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)
}

# Utilize package-included image folder and output to temporary directory

img_dir <- system.file("extdata", "Degenerate_Images", package = "AiES")
svm_model <- system.file("extdata", "svm_example_model.svm", package = "AiES")
axQnt(imprtImg = TRUE, svm_model_path = svm_model, input_dirs = img_dir, output_dir = tempdir())

# Process with custom image resize and background threshold
img_dir <- system.file("extdata", "Intact_Images", package = "AiES")
svm_model <- system.file("extdata", "svm_example_model.svm", package = "AiES")
axQnt(imprtImg = TRUE, resizeW = 700, subBack = 50,
svm_model_path = svm_model, input_dirs = img_dir, output_dir = tempdir())


# Utilize package-included precomputed feature data from .txt files
# and output to temporary directory
txt_dir <- system.file("extdata", "Degenerate_txt", package = "AiES")
svm_model <- system.file("extdata", "svm_example_model.svm", package = "AiES")
axQnt(imprtImg = FALSE, svm_model_path = svm_model, 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.

Usage

```
axSvm(
  nCst = 3,
  nGmm = 0.1,
  nCrss = 5,
  degenerate_path = NULL,
  intact_path = NULL,
  output_data_path = tempdir(),
  output_model_path = tempdir()
)
```

Arguments

nCst	Numeric. SVM cost parameter (C-value) controlling margin hardness. Higher values increase model complexity. (default: 3)
nGmm	Numeric. SVM gamma parameter ((γ)-value) controlling RBF kernel width. Smaller values mean larger kernel radius. (default: 0.1)
nCrss	Integer. Number of folds for cross-validation. Recommended values 5-10. (default: 5)
degenerate_path	Character vector. Path to feature data for the "Degenerate" group (folder containing .txt files or direct file path). If NULL, GUI selection dialog will be shown. (default: NULL)
intact_path	Character vector. Path to feature data for the "Intact" group (folder containing .txt files or direct file path). If NULL, GUI selection dialog will be shown. (default: NULL)
output_data_path	Character. Path to save combined feature data (TSV). If NULL, GUI save dialog will be shown. (default: NULL) 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.(default: NULL)
output_model_path	Character. Path to save trained SVM model (Rdata format). If NULL, 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.(default: NULL)

Details

The function implements a complete machine learning pipeline:

1. **Input handling**: If input paths are not specified, the user is prompted to select folders via a GUI dialog (cross-platform support for RStudio, tcltk, svDialogs, or manual input).
2. **Data preprocessing**: The function reads all .txt files in the specified folders, merges data, and labels them as "Degenerate" or "Intact".
3. **Feature selection**: Focuses on 7 key morphological features:
   - m.eccentricity (elliptical eccentricity)
   - s.radius.sd (radial distribution uniformity)
   - h.sva.s2 (Sum Variance: scale=2)
   - h.idm.s1 (Inverse Difference Moment: local homogeneity scale=1)
   - h.sen.s1 (Sum Entropy: structural complexity scale=1)
   - m.majoraxis (major axis length)
4. **Model training and evaluation**: Utilizes the e1071 package with a radial basis function (RBF) kernel. Cross-validation is performed.
5. **Result saving**: Exports the merged feature data and trained SVM model to disk, ensuring unique filenames if necessary.

#' @section File Naming and Overwrite Policy:

- If the output path is an existing file, a unique name with a numeric suffix is generated to avoid overwriting.
- If the output path is an existing directory, the output file will be saved there with a default name (date-based).
- If the directory does not exist, the file will be saved in the current working directory.

All steps are performed automatically, requiring minimal user intervention.

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)

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.

Note

- Compatible with feature data from axDistmap
- Output files include timestamp in ISO 8601 format (YYYY-MM-DD)
- Model files can be reloaded using base::load()

Examples

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


# Direct path specification

deg_dir <- system.file("extdata", "Degenerate_txt", package = "AiES")
int_dir <- system.file("extdata", "Intact_txt", package = "AiES")
axSvm(degenerate_path = deg_dir, intact_path = int_dir,
      output_data_path = file.path(tempdir(), "svm_test_data.txt"),
      output_model_path = file.path(tempdir(), "svm_test_model.svm"))

# Custom hyperparameters
deg_dir <- system.file("extdata", "Degenerate_txt", package = "AiES")
int_dir <- system.file("extdata", "Intact_txt", package = "AiES")
axSvm(degenerate_path = deg_dir, intact_path = int_dir,
     nCst = 5, nGmm = 0.05, nCrss = 10,
```

```
  output_data_path = tempdir(),       # specify directory only
  output_model_path = tempdir()       # specify directory only
)


# Specify only output directory; default file names will be used
deg_dir <- system.file("extdata", "Degenerate_txt", package = "AiES")
int_dir <- system.file("extdata", "Intact_txt", package = "AiES")
axSvm(degenerate_path = deg_dir, intact_path = int_dir,
  output_data_path = tempdir(),       # specify directory only
  output_model_path = tempdir()       # specify directory only
)

# In this case, output files will be saved as:
#   results/YYYY-MM-DD_Extracted_data_for_ML.txt
#   results/YYYY-MM-DD_AxClassifier.svm
# (YYYY-MM-DD is the current date)
```

Index