Package: FastSegmentation (via r-universe)

June 9, 2026

Title Unsupervised Cell Segmentation by Fast Gaussian Processes

Version 0.0.1

Author Blythe King [aut, cre], Haoran Yan [aut], Laura Baracaldo [aut], Mengyang Gu [aut]

Maintainer Blythe King `<blyking@gmail.com>`

Description Performs fast Gaussian process-based segmentation of microscopy images using spatial smoothing and data-driven thresholding. Code based on Baracaldo, L., King, B., Yan, H., Lin, Y., Miolane, N., & Gu, M. (2025). ``Unsupervised cell segmentation by fast Gaussian processes.'' arXiv preprint <doi:10.48550/arXiv.2505.18902>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Imports EBImage, stats, magick, pracma, RobustGaSP

Suggests plot3D

NeedsCompilation no

Config/pak/sysreqs cmake libfftw3-dev make libmagick++-dev gsfonts libjpeg-dev libpng-dev libtiff-dev libuv1-dev libssl-dev

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

Date/Publication 2026-06-08 17:40:07 UTC

RemoteUrl https://github.com/cran/FastSegmentation

RemoteRef HEAD

RemoteSha c3cacb7c943a5c9a5fb30aa63a802147a1dd18ec

Contents

`criterion_1` *Determining optimal threshold by criterion 1*

Description

Estimates the optimal threshold using criterion 1.

Usage

```
criterion_1(predmean_mat, delta = 0.01, nugget = TRUE)
```

Arguments

`predmean_mat`	Predictive mean matrix of image
`delta`	Step size for percentages to be tested for criterion 1; default is 0.01
`nugget`	boolean to estimate nugget in robust GaSP model; default is TRUE

Value

Returns a list containing:

`thresholded_image`	Binary matrix after applying optimal threshold
`pixel_counts`	Sum of foreground pixels detected for each percentage threshold
`diff_pixel_counts`	Absolute difference in binary matrix between consecutive percentage thresholds
`grad_mag`	Gradient magnitude.
`estimated_percentage`	Estimated optimal threshold by criterion 1

`eliminate_small_areas` *Eliminating Noise and Small Foreign Object Masks from Cell Mask Matrix*

Description

Filters object masks from from noise or foreign objects that are significantly smaller than the estimated cell size. All objects smaller than a certain threshold based on the mean mask size are removed.

Usage

```
eliminate_small_areas(
  GP_masks,
  middle_threshold = 0.15,
  boundary_threshold = 0.05
)
```

Arguments

`GP_masks`	Cell mask matrix
`middle_threshold`	Size threshold for filtering objects not touching the image boundary (removes anything smaller than threshold * mean mask size); default is 0.15
`boundary_threshold`	Size threshold for filtering objects touching the image boundary (removes anything smaller than threshold * mean mask size); default is 0.05

Value

Returns a cell mask matrix with masks from small foreign objects or noise removed

---

`generate_GP_Masks` *Generate cell masks by fast Gaussian processes*

Description

Generates object masks for cell microscopy images using fast Gaussian processes for smoothing, a data-driven threshold for foreground/background segmentation, and watershed for separating touching cell objects. Note that this function divides the original image into sections for more robust processing.

Usage

```
generate_GP_Masks(
  file_path,
  delta = 0.01,
  nugget = TRUE,
  middle_threshold = 0.15,
  boundary_threshold = 0.05,
  compress_output = FALSE,
  return_gradient = FALSE,
  seed = NULL
)
```

Arguments

`file_path`	File path for cell image to segment
`delta`	Step size for percentages to be tested for criterion 1 (default is 0.01)
`nugget`	Nugget boolean for rgasp()
`middle_threshold`	Size threshold for filtering object masks not touching the image boundary (removes anything smaller than threshold * mean mask size); default is 0.15
`boundary_threshold`	Size threshold for filtering object masks touching the image boundary (removes anything smaller than threshold * mean mask size); default is 0.05
`compress_output`	Determines if binary and cell mask results should be represented in matrix (FALSE) or list (TRUE) form; default is FALSE
`return_gradient`	Returns vertical, horizontal, and magnitude gradients for predictive mean if TRUE; default is FALSE
`seed`	Seed for reproducibility (default is NULL)

Value

Returns a list containing:

`ori_images`	Original version of sectioned image
`processed_images`	Predictive mean for each image section
`gradients`	Vertical, horizontal, and magnitude gradients for predictive mean if `return_gradient` is TRUE
`crit_1_opt_thresholds`	Optimal threshold by criterion 1 for each image section
`connected_parts_count`	Number of unique objects for each image section after thresholding before watershed
`outliers`	IDs for thresholded image section that contains unusually large object counts

`combined_predmean`	Predictive mean matrix for entire image
`combined_thresholded1`	Binary matrix for entire image before watershed
`GP_masks`	Cell masks for entire image after thresholding binary matrix, with each cell mask having a unique ID

Examples

```
# Example: Segmentation of JPEG nuclear channel image
img_path <- system.file("extdata", "example_cells_small.jpg", package = "FastSegmentation")
gp_masks_result <- generate_GP_Masks(img_path)
```

`get_proportion` *Dynamically determine row and column proportions*

Description

Decides how many sections the original image should be split into before processing. A target sub-image size can be specified, and this function will return how many divisions should be made in one dimension.

Usage

```
get_proportion(size, target_min = 200, target_max = 400)
```

Arguments

`size`	Row or column length
`target_min`	Minimum sub-image size; default is 200
`target_max`	Maximum sub-image size; default is 400

Value

Fraction that image section should take up of dimension; if no suitable fraction is determined, 1/4 is selected by default

---

`matrix_which` *Get coordinate information for matrices*

Description

Compresses matrix outputs into a data frame containing coordinates for each value. This is used for compressing the binary and cell mask outputs in the generate_GP_Masks function.

Usage

```
matrix_which(mat)
```

Arguments

`mat`	Numeric matrix

Value

Data frame containing each 2D coordinate and corresponding value

---

`predict_separable_GP` *Generating predictive mean from image matrix*

Description

Generates the predictive mean for an image matrix using a Gaussian process.

Usage

```
predict_separable_GP(output_mat, parameters, seed = NULL)
```

Arguments

`output_mat`	Image matrix
`parameters`	Range and nugget parameters estimated using `separable_GP_param_est()`
`seed`	Seed for reproducibility (default is NULL)

Value

Returns a list containing:

`predmean_mat`	Predictive mean matrix
`grad1`	Horizontal gradient
`grad2`	Vertical gradient
`grad_magnitude`	Gradient magnitude
`param`	Gaussian parameters

Examples

```
# File path to TIF, PNG, or JPEG image
library(magick)
img_path <- system.file("extdata", "example_cells.jpg", package = "FastSegmentation")
img <- image_read(img_path)
img_matrix <- as.numeric(img[[1]])[1:100,1:100,1]

# GP parameter estimation
img_gp_parameters <- separable_GP_param_est(img_matrix)

# Predictive mean and gradient calculation
predmean_results <- predict_separable_GP(img_matrix, img_gp_parameters$param)
```

`separable_GP_param_est` *Estimate nugget and range parameters for predictive mean of an image matrix*

Description

Estimates the nugget and range parameters of the Gaussian process for constructing the predictive mean of the segmented paper via Eigendecomposition.

Usage

```
separable_GP_param_est(output_mat, seed = NULL)
```

Arguments

`output_mat`	Image matrix
`seed`	Seed for reproducibility (default is NULL)

Value

Returns a list containing the range and nugget parameters

`threshold_image` *Setting a threshold to create a binary image matrix*

Description

Sets a threshold based on quantile of pixel value to create a binary image matrix, where foreground pixels correspond with cell objects.

Usage

```
threshold_image(mat, percentage, count = TRUE)
```

Arguments

`mat`	Image matrix
`percentage`	Percentage cutoff for foreground and background pixels (estimated using `criterion_1()`)
`count`	Boolean to calculate sum of foreground pixels after thresholding; default is TRUE

Value

Returns either the sum of foreground pixels (when count is TRUE) or the binary matrix after thresholding (when count is FALSE)

Index