

Package ‘morph’

October 5, 2026

Type Package

Title 3D Segmentation of Voxels into Morphologic Classes

Version 2.0.0

Date 2026-10-05

Maintainer Tarmo K. Remmel <remmelt@yorku.ca>

Description Automatically segments a 3D array that represents a volume of binary voxels into mutually exclusive morphological elements. This package extends existing work for segmenting 2D binary raster data. A paper documenting this approach has been published in the journal Landscape Ecology: Remmel, T.K. (2022) <[doi:10.1007/s10980-021-01384-7](https://doi.org/10.1007/s10980-021-01384-7)>. The output is a cartridge (list object) that maintains the input array, the segmentation results in array format, and a summary table. Plotting functionality is provided to produce interactive visual outputs from the produced results cartridge, allowing custom plotting to be performed separately from the segmentation, which speeds-up processing. While the old functions persist, they are being phased out and will eventually be replaced with the new run-morph3d() and plotmorph3d() functions.

License GPL-3

LazyData true

LazyDataCompression xz

Depends R (>= 4.1.0), rgl, reshape2, igraph, stringr

NeedsCompilation no

Repository CRAN

Author Tarmo K. Remmel [aut, cre] (ORCID:
<<https://orcid.org/0000-0001-6251-876X>>)

Date/Publication 2026-10-05 15:00:02 UTC

Contents

morph-package	2
arraytrim	4
LEdemo	5
morph3d	6

morph3dlinks	8
morph3dplot	9
morph3dprep	10
plotmorph3d	11
runmorph3d	13
Index	16

morph-package	<i>3D Segmentation of Voxels into Morphologic Classes</i>
---------------	---

Description

Automatically segments a 3D array that represents a volume of binary voxels into mutually exclusive morphological elements. This package extends existing work for segmenting 2D binary raster data. A paper documenting this approach has been published in the journal Landscape Ecology: Remmel, T.K. (2022) <doi:10.1007/s10980-021-01384-7>. The output is a cartridge (list object) that maintains the input array, the segmentation results in array format, and a summary table. Plotting functionality is provided to produce interactive visual outputs from the produced results cartridge, allowing custom plotting to be performed separately from the segmentation, which speeds-up processing. While the old functions persist, they are being phased out and will eventually be replaced with the new runmorph3d() and plotmorph3d() functions.

Details

The DESCRIPTION file:

Package:	morph
Type:	Package
Title:	3D Segmentation of Voxels into Morphologic Classes
Version:	2.0.0
Date:	2026-10-05
Authors@R:	c(person("Tarmo K.", "Remmel", email = "remmelt@yorku.ca", role = c("aut", "cre"), comment =
Maintainer:	Tarmo K. Remmel <remmelt@yorku.ca>
Description:	Automatically segments a 3D array that represents a volume of binary voxels into mutually exclusiv
License:	GPL-3
LazyData:	true
LazyDataCompression:	xz
Depends:	R (>= 4.1.0), rgl, reshape2, igraph, stringr
NeedsCompilation:	no
Repository:	CRAN
Packaged:	2026-10-05 14:03:00 UTC+3
Author:	Tarmo K. Remmel [aut, cre] (< https://orcid.org/0000-0001-6251-876X >)
Date/Publication:	2026-10-05 14:03:00 UTC+3

Index of help topics:

LEdemo	Landscape Ecology Demo Data (3D Voxels)
arraytrim	Trim excess white-space (0 planes) from a 3D binary array.
morph-package	3D Segmentation of Voxels into Morphologic Classes
morph3d	Performs a morphological segmentation of a binary 3D array
morph3dlinks	An internal function that computes the spatial neighbours for voxels. It is an internal function called by morph3d() and not intended to be called on its own.
morph3dplot	A plot function that converts a 3D array of voxels into an interactive rgl plot.
morph3dprep	Prepare 3D voxel data for interactive 3D plotting.
plotmorph3d	A plotting function for morphological segmentation results.
runmorph3d	Performs a morphological segmentation of a binary 3D array

This package handles the segmentation of binary voxel data, provided as a numeric 3D array, into its constituent morphological classes. Each voxel is assigned to one of nine mutually exclusive morphological classes: OUTSIDE, MASS, SKIN, CRUMB, CIRCUIT, ANTENNA, BOND, VOID-SKIN, or VOID-VOLUME.

The current version of morph 2.0.0 is a substantial update to the original release. While the old functions remain, their use is discouraged. The new version is simpler, cleaner, more adaptable, but maintains consistency in result codes produced. The intent is to retire the original functions in the next release, and to have the current functions persist. The first function is runmorph3d() which processes an input 3D array (representing voxels), to produce a 3D morphological segmentation. The results are stored in a standardized cartridge (a list of objects) that can subsequently be passed to plotmorph3d() to handle a wide range of plotting functionality.

Author(s)

Tarmo K. Remmel [aut, cre] (<<https://orcid.org/0000-0001-6251-876X>>) Maintainer: Tarmo K. Remmel <remmelt@yorku.ca>

References

Remmel, T.K. 2022. Extending morphological pattern analysis to 3D voxels. *Landscape Ecology* 37(2):373-380.

See Also

See Also [melt](#)

Examples

Need to provide some simple examples here.

arraytrim

Trim excess white-space (0 planes) from a 3D binary array.

Description

A helper function for reducing the size of 3D arrays if necessary. This reduction can improve memory utilization and speed processing, since voxels outside the feature of interest are eliminated.

Usage

```
arraytrim(VOLOBJ = NULL)
```

Arguments

VOLOBJ	This is a 3D binary (0,1) numeric array, which represent voxels. If there are outer margins that are all zeros, they will be trimmed off to reduce the size of the array.
--------	---

Details

The function scans inward, starting at the outer margins, for planes that are all zeros. If they are, those planes will be trimmed at the end. Only planes of zeros that connect contiguously to the outer margins will be trimmed.

Value

The output is an array with the excess outer zero planes removed.

Note

It can happen that trimming reduces the resulting dimensions to be <3, and thus, prior to running morph3d(), ensure that your data still fits the requirements.

Author(s)

Tarmo K. Remmel

References

Remmel, T.K. 2022. Extending morphological pattern analysis to 3D voxels. *Landscape Ecology* 37(2):373-380.

See Also

See Also [morph3d](#), [morph3dprep](#)

Examples

```
LEdemoSmall <- arraytrim(LEdemo)
```

LEdemo

Landscape Ecology Demo Data (3D Voxels)

Description

This is a 3D numeric array with dimensions [10,10,10], representing 1000 voxels for demonstration purposes. The data was manually produced to ensure that all morphological classes are present, to allow demonstration and illustration within the Landscape Ecology paper in which the `morph3d()` functionality was introduced.

Usage

```
data("LEdemo")
```

Format

The format is: `num [1:10, 1:10, 1:10] 1 1 0 0 0 0 0 0 0 ...`

Details

This data can be used for testing, demonstrations, or for quickly getting the package up and running, before getting your own data into the proper format.

Source

This data was produced by the author of the package for demonstration purposes and was used in the paper referenced below as the example data.

References

Rommel, T.K. 2022. Extending morphological pattern analysis to 3D voxels. *Landscape Ecology* 37(2):373-380.

Examples

```
data(LEdemo)
str(LEdemo)
```

morph3d

*Performs a morphological segmentation of a binary 3D array***Description**

This is the extension of 2D morphological segmentation of binary pixels in a planar raster setting to the 3D domain of voxels. A 3D binary array with 1 (belonging to the feature of interest) and 0 (not belonging to the feature of interest) is the only input required for the mathematical morphological segmentation to proceed. The result is a list object containing the original input data, the categorical segmentation layer, and various summary outputs that may be necessary at later interpretation stages. The algorithm assigns voxels into mutually exclusive categories: outside, mass, skin, crumb, circuit, antenna, bond, void-volume, and void based on their relative positioning and connectivity within the feature of interest.

Usage

```
morph3d(DATA_CUBE = NULL, VERBOSE = FALSE, PLOT = FALSE, FINALPLOT = TRUE, PLOTIDS = FALSE)
```

Arguments

DATA_CUBE	This is the input 3D data array. The dimensions do not need to be equal, but the data must be binary (0,1) and be of numeric type.
VERBOSE	This is a Boolean flag that turns on off additional feedback as the code is running. This is generally used for testing and debugging and can easily be set to FALSE with no concern over how the code runs.
PLOT	This is a Boolean flag that turns on off 3D interactive environment plots for each mutually exclusive morphological class. If these plots are not explicitly required, set this to FALSE, as the plotting is the bottleneck that dramatically slows processing.
FINALPLOT	This is a Boolean flag that turns on off the 3D interactive environment plot for the final voxel segmentation result. If this is set to FALSE, it can readily be plotted in the future, since the results are saved by the function to a compound list object and the plotting function is supplied with this package.
PLOTIDS	This is a Boolean flag that turns on off the plotting of unique voxel IDS at the centroids of each voxel during plotting. These can be very useful for identifying specific voxels and associations, but can also clutter the view if they are not expressly required.

Details

Each voxel can belong to only a single 3D morphological class. The concept expands the planar morphological segmentation available in the MSPA software package and establishes corresponding terminology to distinguish the 3D case. The processing logic is presented and a fully-functional environment allows any binary 3D data to be processed with this engine. This tool is currently limited by dataset size, where processing can take a very long time and produce strange artifacts. This is something that is actively being worked on and a Python library has been released to offer

alternatives and mitigate some of this concern. I hope to optimize this R package for a new release in the future that alleviates this limitation.

Value

The function returns a LIST object with the following items:

OriginalData	This is the original input 3D data array of 0,1 values.
Graph	This is the network graph object in igraph format for the feature of interest.
VoxelIDs	This is a 3D array with the same dimensions as the input data and contains the unique voxel ID value for each voxel.
ObjectID	If there are multiple disconnected components for the input feature of interest, each component is assigned a unique object ID. This 3D array with the same dimensions as the input data contains those unique object ID values for each voxel.
Morphology	This is a 3D array with the same dimensions as the input data and is a categorical voxel map identifying which morphological class each voxel belongs to. The codes are: 1=outside, 2=mass, 3=skin, 4=crumb, 5=circuit, 6=antenna, 7=bond, 8=void-volume, and 9=void.
Cores	This is a 3D array with the same dimensions as the input data and contains the codes for unique cores (of mass voxels) that may exist within the feature of interest.
ExpCores	This is the same as Cores, but expanded outward to enlarge the volume represented by each core. This is used for internal processing but may be of use or benefit to some, so it is also provided here.
Summary	This is a tabular summary of the morphological segmentation by class. The call name, number of voxels, and the percentage of the total volume is summarized for the actual data provided in the Morphology item.
Egg	This is a temporary output that is used for checking and debugging. It is likely to be eliminated in the future.
Bgrnd	This 3D array identifies the voxels outside the feature of interest, but that are not part of the voids internal to the structure itself.
VOIDvolume	This 3D array identifies the voxels that form the holes, gaps, or voids within the structure.

Note

Current implementation considers voxels as neighbours if they share a face only (orthogonal neighbours). Expansion to include diagonal neighbours, where voxels share an edge or a point, are being considered. The use of rgl provides functionality to produce interactive 3D plots that greatly facilitate visualization, but dramatically slow the processing as input volumes increase in size. Avoid plotting unless explicitly required. You are able to plot from the stored list object at anytime afterwards too, it is not necessary to plot during the segmentation call.

Author(s)

Tarmo K. Rimmel

References

Rommel, T.K. 2022. Extending morphological pattern analysis to 3D voxels. *Landscape Ecology* 37(2):373-380. [DOI: 10.1007/s10980-021-01384-7]

See Also

See Also [morph3dplot](#), [morph3dprep](#)

Examples

```
# SET FINALPLOT=TRUE IF THE INTERACTIVE 3D PLOT IS DESIRED
# THAT CONTAINS THE FINAL SEGMENTATION CATEGORIES
# IF A PLOT FOR EACH CATEGORY IS REQUIRED, SET PLOT=TRUE
LEdemoOut <- morph3d(LEdemo, PLOT=FALSE, FINALPLOT=FALSE)
str(LEdemoOut)
```

morph3dlinks	<i>An internal function that computes the spatial neighbours for voxels. It is an internal function called by morph3d() and not intended to be called on its own.</i>
--------------	---

Description

An internal helper function to compute voxel neighbours in orthogonal 3D space and to return the information in data frame format back to morph3d(), the function that calls this function.

Usage

```
morph3dlinks(VOLOBJ = NULL, VOXELIDS = NULL, VERBOSE = FALSE)
```

Arguments

VOLOBJ	This is a 3D array that is larger in all dimensions, within which the actual data array is inset. This ensures that outer margins will be properly processed.
VOXELIDS	This is a 3D array of the unique voxel IDS, which are used to track individual voxels and their connectivity.
VERBOSE	This is a Boolean flag for turning on off additional feedback during processing.

Details

This internal function simply makes the total length of the morph3d() function shorter.

Value

This function returns an intermediate data frame of voxel neighbours that is subsequently pared down as morph3d() continues.

Note

No additional notes at this time.

Author(s)

Tarmo K. Remmel

References

Remmel, T.K. 2022. Extending morphological pattern analysis to 3D voxels. *Landscape Ecology* 37(2):373-380.

See Also

See Also [morph3d](#)

Examples

```
# THIS IS AN INTERNAL FUNCITON AND USAGE EXAMPLES ARE NOT NECESSARY
```

morph3dplot	<i>A plot function that converts a 3D array of voxels into an interactive rgl plot.</i>
-------------	---

Description

This plot functionality can be used for any 3D array data that represent voxels. This functionality is called directly by morph3d() but can be called independently by passing it 3D array data that has been processed by morph3dprep().

Usage

```
morph3dplot(data=NULL, CELLID=FALSE, LEGEND=FALSE, ORIGTRANSP=TRUE, CELLLABELS=NULL)
```

Arguments

data	This is the voxel data to be plotted and needs to be processed with morph3dprep() first.
CELLID	A Boolean flag used to control whether the unique voxel IDs are plotted or not.
LEGEND	A Boolean flag that determines whether a legend will be drawn on the output interactive plot or not. When this is set to TRUE, plotting is much slower.
ORIGTRANSP	A Boolean flag that is used to set the transparency of original binary input voxel data. It can be used to help produce better visualizations but is not an overly important argument for most.
CELLLABELS	A 3D array with values to be used for labelling individual voxels in the plot. These will only plot if CELLID = TRUE.

Details

Output classes are coloured (with varying degrees of transparency to make classes easier to see) as follows. Mass = "green", Skin = "black", Crumb = "brown", Circuit = "orange", Antenna = "pink", Bond = "cornflowerblue", Void-volume = "navy", and Void = "seagreen".

Value

The output is an interactive 3D rgl plot of the voxels supplied as input. Morphological segmentations produced by morph3d() will always have elements drawn in consistent color for ease of interpretation.

Note

No further notes available at this time.

Author(s)

Tarmo K. Remmel

References

Remmel, T.K. 2022. Extending morphological pattern analysis to 3D voxels. *Landscape Ecology* 37(2):373-380.

See Also

See Also [morph3d](#), [morph3dprep](#)

Examples

```
LEdemoOut <- morph3d(DATAcube = LEdemo, VERBOSE = FALSE, FINALPLOT = FALSE)

# PLOTTING IS TIME INTENSIVE AND THESE FUNCTIONS ARE BEING RETIRED SHORTLY
# BEST TO LOOK AT THE NEW runmorph3d() and plotmorph3d() FUNCTIONS WHICH RUN MORE RAPIDLY
morph2plot <- morph3dprep(incube=LEdemoOut$Morphology, ORIG=FALSE)
morph3dplot(data=morph2plot, CELLID=FALSE, LEGEND=FALSE, ORIGTRANSP=TRUE)
```

morph3dprep

Prepare 3D voxel data for interactive 3D plotting.

Description

Prepares 3D array data for plotting.

Usage

```
morph3dprep(incube = NULL, orig = FALSE)
```

Arguments

INCUBE	This is the input 3D array of voxels.
ORIG	A Boolean flag to control whether the data is the original input binary data (TRUE), or some other morphological segmentation (FALSE).

Details

No additional details at this time.

Value

The output is...

Note

No additional notes at this time.

Author(s)

Tarmo K. Remmel

References

Remmel, T.K. 2022. Extending morphological pattern analysis to 3D voxels. *Landscape Ecology* 37(2):373-380.

See Also

See Also [morph3d](#), [morph3dplot](#)

Examples

```
out <- morph3dprep(INCUBE = LEdemo, ORIG = FALSE)
print(out)
```

plotmorph3d

A plotting function for morphological segmentation results.

Description

This plot function is intended to be used with a morphological segmentation produced by `run-morph3d()` and the corresponding cartridge that contains the results (a standardized list object). The function offers many arguments that permit control over what is plotted, in what colours, with how much transparency, whether voxel boundaries are to be drawn and so on. The intent is that this plotting function runs completely independently of the segmentation function and that it can be run on cartridges at any time, rather than relying on plotting only during the segmentation process. This offers greater flexibility and speed. The default is to use standardised colours and display settings for viewing all morphological codes simultaneously on an interactive 3D output viewer.

Usage

```
plotmorph3d(CARTRIDGE = NULL, colours = c("red", "green",
    "darkgrey", "brown", "orange", "pink", "cornflowerblue", "navy",
    "seagreen"), alphas = c(0, 0.65, 0.65, 0.65, 0.65, 0.9, 0.65,
    0.65, 0.65), voxel_size = 1, show_axes = TRUE, show_box = TRUE,
    show_wireframe = FALSE, show_labels = FALSE, label_array = NULL,
    wireframe_colour = "black", wireframe_alpha = 0.5, label_colour = "black",
    label_cex = 1)
```

Arguments

CARTRIDGE	This is the voxel data to be plotted and needs to be produced with morph3dprep() first.
colours	An optional vector of colours that are to be used for controlling plot colours for specific categories.
alphas	An optional vector of identical length to colours, providing an alpha level (transparency, between 0 and 1) for each colour.
voxel_size	A number that controls voxel size in plots. The default is set to 1; likely best to leave it at that until you have a reason to change it.
show_axes	A Boolean control on whether axes are plotted or not.
show_box	A Boolean control on whether the bounding box is plotted or not.
show_wireframe	A Boolean control on whether voxel outlines (the wireframe) are plotted or not.
show_labels	A Boolean control on whether voxel labels are plotted or not.
label_array	If show_labels = TRUE, then provide a 3D array of the same dimensions as the morphological segmentation result with voxel values to be used as voxel labels. The default is to use the \$MORPHCODE morphological segmentation results.
wireframe_colour	An option to include a colour for drawing the wireframe.
wireframe_alpha	An optional numeric value controlling the transparency of the wireframe.
label_colour	An optional opportunity to control label colour.
label_cex	An optional opportunity to control the size of label text.

Details

Output classes are coloured (with varying degrees of transparency to make classes easier to see) as follows. MASS = "green", SKIN = "darkgrey", CRUMB = "brown", CIRCUIT = "orange", ANTENNA = "pink", BOND = "cornflowerblue", VOID-SKIN = "navy", and VOID-VOLUME = "seagreen".

Value

The output is an interactive 3D rgl plot of the voxels supplied as input. Morphological segmentations produced by runmorph3d() will always have elements drawn in consistent color for ease of interpretation.

Note

No further notes available at this time.

Author(s)

Tarmo K. Remmel

References

Remmel, T.K. 2022. Extending morphological pattern analysis to 3D voxels. *Landscape Ecology* 37(2):373-380.

See Also

See Also [runmorph3d](#)

Examples

```
# PERFORM A 3D MORPHOLOGICAL SEGMENTATION TO PRODUCE A RESULT CARTRIDGE
result <- runmorph3d(LEdemo)

# CALL THE PLOT FUNCTION
plotmorph3d(result$MORPHCODE)
```

runmorph3d

Performs a morphological segmentation of a binary 3D array

Description

This is the extension of 2D morphological segmentation of binary pixels in a planar raster setting to the 3D domain of voxels. A 3D binary array with 1 (belonging to the feature of interest) and 0 (not belonging to the feature of interest) is the only input required for the mathematical morphological segmentation to proceed. The result is a list object containing the original input data (a results cartridge), the categorical segmentation layer, and various summary outputs that may be necessary at later interpretation stages. The algorithm assigns voxels into mutually exclusive categories: OUTSIDE (1), MASS (2), SKIN (3), CRUMB (4), CIRCUIT (5), ANTENNA (6), BOND (7), VOID-VOLUME (8), and VOID-SKIN (9) based on their relative positioning and connectivity within the feature of interest.

Usage

```
runmorph3d(DATA = NULL)
```

Arguments

DATA	This is the input 3D data array. The dimensions do not need to be equal, but the data must be binary integers (0,1).
------	--

Details

Each voxel can belong to only a single 3D morphological class. The concept expands the planar morphological segmentation available in the MSPA software package and establishes corresponding terminology to distinguish the 3D case. The processing logic is presented and a fully-functional environment allows any binary 3D data to be processed with this engine. This tool is currently limited by dataset size, where processing can take a very long time and produce strange artifacts. Once the segmentation is complete, all plotting options are performed with the function `plotmorph3d`

Value

The function returns a LIST object with the following items:

DATE	The date and time that the morphological segmentation was performed.
NOTE	A simple character string stating "3D Morphological Segmentation" to identify the contents of the cartridge.
INPUTDATA	This is the original input 3D data array of 0,1 values.
SUMMARY	This is a data.frame summary of the morphological segmentation by class. The classification number (as an integer), the class name, the number of voxels, and the percentage of the total volume is summarized for the actual data processed.
MORPHCODE	This is a 3D array containing the morphological code assigned to each voxel. This is the spatial result that corresponds to the tabular SUMMARY above.

Note

Current implementation considers voxels as neighbours if they share a face only (orthogonal neighbours). Expansion to include diagonal neighbours, where voxels share an edge or a point, are being considered. The use of `rgl` provides functionality to produce interactive 3D plots that greatly facilitate visualization, but dramatically slow the processing as input volumes increase in size. Avoid plotting unless explicitly required. You are able to plot from the stored list object at anytime afterwards too, it is not necessary to plot during the segmentation call.

Author(s)

Tarmo K. Remmel

References

Remmel, T.K. 2022. Extending morphological pattern analysis to 3D voxels. *Landscape Ecology* 37(2):373-380. DOI: 10.1007/s10980-021-01384-7

See Also

See Also [plotmorph3d](#)

Examples

```
# PERFORM THE 3D SEGMENTATION ON THE DEMO DATA OBJECT
result <- runmorph3d(DATA=LEdemo)
# LOOK AT THE STRUCTURE OF THE CARTRIDGE PRODUCED
str(result)
# VIEW THE TABULAR SUMMARY
result$SUMMARY
```

Index

- * **array**
 - arraytrim, [4](#)
 - morph3d, [6](#)
 - morph3dplot, [9](#)
 - plotmorph3d, [11](#)
 - runmorph3d, [13](#)
- * **datasets**
 - LEdemo, [5](#)
- * **iplot**
 - morph-package, [2](#)
 - morph3dplot, [9](#)
 - plotmorph3d, [11](#)
- * **manip**
 - morph-package, [2](#)
 - morph3d, [6](#)
 - morph3dlinks, [8](#)
 - morph3dprep, [10](#)
 - runmorph3d, [13](#)
- * **package**
 - morph-package, [2](#)

arraytrim, [4](#)

LEdemo, [5](#)

melt, [3](#)

morph (morph-package), [2](#)

morph-package, [2](#)

morph3d, [4](#), [6](#), [9–11](#)

morph3dlinks, [8](#)

morph3dplot, [8](#), [9](#), [11](#)

morph3dprep, [4](#), [8](#), [10](#), [10](#)

plotmorph3d, [11](#), [14](#)

runmorph3d, [13](#), [13](#)