Package 'mapscape'

May 4, 2026

Title mapscape

Version 1.37.0

Description MapScape integrates clonal prevalence, clonal hierarchy, anatomic and mutational information to provide interactive visualization of spatial clonal evolution. There are four inputs to MapScape: (i) the clonal phylogeny, (ii) clonal prevalences, (iii) an image reference, which may be a medical image or drawing and (iv) pixel locations for each sample on the referenced image. Optionally, MapScape can accept a data table of mutations for each clone and their variant allele frequencies in each sample. The output of MapScape consists of a cropped anatomical image surrounded by two representations of each tumour sample. The first, a cellular aggregate, visually displays the prevalence of each clone. The second shows a skeleton of the clonal phylogeny while highlighting only those clones present in the sample. Together, these representations enable the analyst to visualize the distribution of clones throughout anatomic space.

Depends R (>= 3.3)

Imports htmlwidgets (>= 0.5), jsonlite (>= 0.9.19), base64enc (>= 0.1-3), stringr (>= 1.0.0)

License GPL-3

LazyData true

RoxygenNote 6.0.1

biocViews Visualization

Suggests knitr, rmarkdown

VignetteBuilder knitr

git_url https://git.bioconductor.org/packages/mapscape

git_branch devel

git_last_commit 7067baf

git_last_commit_date 2026-04-28

Repository Bioconductor 3.24

Date/Publication 2026-05-03

Author Maia Smith [aut, cre]

Maintainer Maia Smith <maiaannesmith@gmail.com>

Contents

mapscape 2
mapscapeOutput 6

Index **7**

`mapscape` *MapScape*

Description

MapScape is a visualization tool for spatial clonal evolution. MapScape displays a cropped anatomical image surrounded by two representations of each tumour sample representing the distribution of clones throughout anatomic space. The first, a cellular aggregate or donut view, displays the prevalence of each clone. The second shows a skeleton of the patient's clonal phylogeny while highlighting only those clones present in the sample. Note: the cellular aggregate does not accurately represent the positions of clones within a sample. We therefore provide the alternative donut chart view as a less artistic representation of the tumour sample. See the Interactivity section below for instructions to switch between views.

Usage

```
mapscape(clonal_prev, tree_edges, sample_locations, img_ref,
  clone_colours = "NA", mutations = "NA", sample_ids = c("NA"),
  n_cells = 100, show_low_prev_gtypes = FALSE,
  phylogeny_title = "Clonal Phylogeny", anatomy_title = "Anatomy",
  classification_title = "Phylogenetic Classification",
  show_warnings = TRUE, width = 960, height = 960)
```

Arguments

`clonal_prev` `data.frame` Clonal prevalence. Required columns are:

sample_id: `character()` id for the tumour sample.

clone_id: `character()` clone id.

clonal_prev: `numeric()` clonal prevalence.

`tree_edges` `data.frame` Tree edges of a rooted tree. Required columns are:

source: `character()` source node id.

target: `character()` target node id.

`sample_locations`

`data.frame` Anatomic locations for each tumour sample. Required columns are:

sample_id: `character()` id for the tumour sample.

location_id: `character()` name of anatomic location for this tumour sample.

x: `numeric()` x-coordinate (in pixels) for anatomic location on anatomic image.

y: `numeric()` y-coordinate (in pixels) for anatomic location on anatomic image.

`img_ref` `character()` A reference for the custom anatomical image to use, *** in PNG format ***, either a URL to an image hosted online or a path to the image in local file system.

`clone_colours` `data.frame` (Optional) Clone ids and their corresponding colours (in hex format). Required columns are:

clone_id: `character()` clone id.

colour: `character()` the corresponding Hex colour for each clone id.

`mutations` `data.frame` (Optional) Mutations occurring at each clone. Any additional field will be shown in the mutation table. Required columns are:

chrom: `character()` chromosome number.

coord: `numeric()` coordinate of mutation on chromosome.

clone_id: `character()` clone id.

sample_id: `character()` id for the tumour sample.

VAF: `numeric()` ariant allele frequency of the mutation in the corresponding sample.

`sample_ids` `vector` (Optional) Ids of the samples in the order your wish to display them (clockwise from positive x-axis).

`n_cells` `numeric()` (Optional) The number of cells to plot (for voronoi tessellation).

`show_low_prev_gtypes` `logical()` (Optional) Whether or not to show low-prevalence (< 0.01) clones in the view. Default is FALSE.

`phylogeny_title` `character()` (Optional) Legend title for the phylogeny. Default is "Clonal Phylogeny".

`anatomy_title` `character()` (Optional) Legend title for the anatomy. Default is "Anatomy".

`classification_title` `character()` (Optional) Legend title for the phylogenetic classification. Default is "Phylogenetic Classification".

`show_warnings` `logical()` (Optional) Whether or not to show any warnings. Default is TRUE.

`width` `numeric()` (Optional) Width of the plot. Minimum width is 930.

`height` `numeric()` (Optional) Height of the plot. Minimum height is 700.

Details

Interactive components in the top toolbar:

1. Click the download buttons to download the current view as PNG or SVG.
2. Click the reset button to exit a clone or mutation selection.
3. Click the view switch button to switch between cellular

Interactive components in main view:

1. Reorder samples by grabbing the sample name or cellular aggregate / donut and dragging it radially.
2. Hover over anatomic location of interest to view the anatomic location name and the patient data associated with that location.
3. Hover over a tree node of a particular sample to view cellular prevalence of that clone in that particular sample.

Interactive components in legend:

1. Hover over legend tree node to view the clone ID as well as the clone's prevalence at each tumour sample. Any anatomic locations expressing that clone will be highlighted.
2. Hover over legend tree branch to view tumour samples expressing all descendant clones.
3. Click on legend tree node(s) to view (a) updated mutations table showing novel mutations at that clone(s), and (b) tumour samples expressing the novel mutations at that clone(s).
4. Hover over a mixture class (e.g. "pure", "polyphyletic", "monophyletic") to view corresponding tumour samples, and the participating phylogeny in each tumour sample.

Interactive components in mutation table:

1. Search for any chromosome, coordinate, gene, etc.
2. Click on a row in the table, and the view will update to show the tumour samples with that mutation, and the variant allele frequency for that mutation in each tumour sample.
3. Sort the table by a column (all columns sortable except the Clone column).

Note: Click on the reset button to exit a selection. Click the download buttons to download a PNG or SVG of the view.

Examples

```
library("mapscape")

# EXAMPLE 1 - Patient A21, Gundem et al., 2015

# clonal prevalences
clonal_prev <- read.csv(system.file("extdata", "A21_clonal_prev.csv",
    package = "mapscape"))

# mutations
mutations <- read.csv(system.file("extdata", "A21_mutations.csv",
    package = "mapscape"))

# locations of each tumour sample on user-provided image
sample_locations <- read.csv(system.file("extdata",
    "A21_sample_locations.csv", package = "mapscape"))

# genotype tree edges
tree_edges <- read.csv(system.file("extdata", "A21_tree.csv", package
    = "mapscape"))

# image reference
img_ref <- system.file("extdata", "A21_anatomical_image.png", package
    = "mapscape")

# radial order of samples
sample_ids <- c("H","F","J","D","A","I","C","E","G")

# run mapscape
mapscape(clonal_prev = clonal_prev, tree_edges = tree_edges,
    sample_locations = sample_locations, mutations = mutations,
img_ref = img_ref, sample_ids = sample_ids)
```

```
# EXAMPLE 2 - Patient 1, McPherson and Roth et al., 2016

# clonal prevalences
clonal_prev <- read.csv(system.file("extdata", "px1_clonal_prev.csv",
    package = "mapscape"))

# mutations
mutations <- read.csv(system.file("extdata", "px1_mutations.csv",
    package = "mapscape"))

# locations of each tumour sample on user-provided image
sample_locations <- read.csv(system.file("extdata",
    "px1_sample_locations.csv", package = "mapscape"))

# genotype tree edges
tree_edges <- read.csv(system.file("extdata", "px1_tree.csv",
    package = "mapscape"))

# image reference
img_ref <- system.file("extdata", "px1_anatomical_image.png",
    package = "mapscape")

# colours for each clone
clone_colours <- data.frame(
    clone_id = c("A","B","C","D","E","F","G","H","I"),
    colour = c("d0ced0", "2CD0AB", "7FE9D1", "FFD94B", "FD8EE5",
    "F8766D", "4FD8FF", "B09AF5", "D4C7FC"))

# radial order of samples
sample_ids <- c("LFTB4", "LOvB2", "ApC1", "ROvA4", "ROv4", "ROv3",
    "ROv2", "ROv1", "RFTA16", "Om1", "SBwl", "SBwlE4")

# run mapscape
mapscape(clonal_prev = clonal_prev, tree_edges = tree_edges,
    sample_locations = sample_locations, mutations = mutations,
    img_ref = img_ref, clone_colours = clone_colours,
    sample_ids = sample_ids)

# EXAMPLE 3 - Patient 7, McPherson and Roth et al., 2016

# clonal prevalences
clonal_prev <- read.csv(system.file("extdata", "px7_clonal_prev.csv",
    package = "mapscape"))

# mutations
mutations <- read.csv(system.file("extdata", "px7_mutations.csv",
    package = "mapscape"))

# locations of each tumour sample on user-provided image
sample_locations <- read.csv(system.file("extdata",
    "px7_sample_locations.csv", package = "mapscape"))

# genotype tree edges
tree_edges <- read.csv(system.file("extdata", "px7_tree.csv",
    package = "mapscape"))

# image reference
```

```
img_ref <- system.file("extdata", "px7_anatomical_image.png",
    package = "mapscape")

# colours for each clone
clone_colours <- data.frame(clone_id = c("A","B","C","D","E"),
                            colour = c("d0ced0", "2CD0AB", "FFD94B",
    "FD8EE5", "F8766D"))

# radial order of samples
sample_ids <- c("BwlA6", "RPvM", "RUtD1", "RUtD2", "RUtD3", "ROvC4",
    "ROvC5", "ROvC6", "LOv1","LOvA10","LOvA4","BrnM", "BrnMA1")

# run mapscape
mapscape(clonal_prev = clonal_prev, tree_edges = tree_edges,
    sample_locations = sample_locations, mutations = mutations,
    img_ref = img_ref, clone_colours = clone_colours, sample_ids = sample_ids)
```

`mapscapeOutput` *Widget output function for use in Shiny*

Description

Widget output function for use in Shiny

Widget render function for use in Shiny

Usage

```
mapscapeOutput(outputId, width = "100%", height = "400px")

renderMapscape(expr, env = parent.frame(), quoted = FALSE)
```

Arguments

`outputId`	– id of output
`width`	– width of output
`height`	– height of output
`expr`	– expression for Shiny
`env`	– environment for Shiny
`quoted`	– default is FALSE

Examples

```
mapscapeOutput(1, '100%', '300px')
mapscapeOutput(1, '80%', '300px')
mapscapeOutput("hi")
mapscapeOutput("hello")
```

Index

mapscape, 2
mapscapeOutput, 6

renderMapscape *(*mapscapeOutput*)*, 6